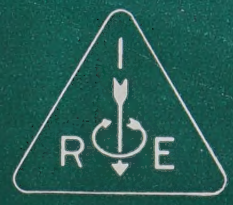
IRE
Transactions
on ELECTRONIC COMPUTERS



Volume EC-8 **SEPTEMBER, 1959** **Number 3**

Published Quarterly

PUBLISHED BY THE
Professional Group on ELECTRONIC COMPUTERS

IRE Transactions® on Electronic Computers

Published by the Institute of Radio Engineers, Inc., for the Professional Group on Electronic Computers at 1 East 79th Street, New York 21, N.Y. Responsibility for the contents rests upon the authors and not upon the IRE, the Group, or its members. Price per copy: IRE-PGEC members, $2.15; IRE members, $3.20, nonmembers, $6.45. Yearly subscriptions rate: nonmembers, $17.00; colleges and public libraries, $12.75. Address requests to The Institute of Radio Engineers, 1 East 79th Street, N.Y. 21, N.Y.

Notice to Authors: Address all papers and editorial correspondence to H. E. Tompkins, The Moore School of Electrical Engineering, 200 South 33rd Street, Philadelphia 4, Pa. To avoid delay and inconvenience in the processing of manuscripts, please follow the procedure suggested on the back cover, under Information for Authors.

The Chairman's Column

To Members of the PGEC

Picking up where Willis Ware left off, I plan to be in touch with you from time to time through this column concerning PGEC operations, both present and planned. But keeping the information pipeline open to the membership is but half the job—and I invite you to keep the pipeline open to the administrative committee. Nominally, the PGEC membership is represented at administrative committee meetings through the committee membership which is elected on a basis of geographical distribution of members, five new members coming into the committee each year for three year terms. I believe that the committee is effective in representation of the membership, but your representation can be greatly improved by: direct communication with me, direct communication with Editor Tompkins, direct communication with any of the administrative committee or standing or Ad Hoc committee chairmen (listed in the June and subsequent issues of these Transactions), or your personal attendance at any of the administrative committee meetings. Incidentally, our next administrative committee meeting will have been held during the WESCON Convention by the time you read this.

Underscoring the importance which the administrative committee places on two areas of thought —namely, the PGEC's position in an international computer societies federation, and a second problem which is always with us, our long-range planning—I have formed two new Ad Hoc committees. Dr. Willis Ware was named chairman of a committee to study our foreign relationships, and Dr. Arnold Cohen, newly elected PGEC Vice-Chairman, was named chairman of a committee on long-range planning. Both of these subjects should be centers of continuing interest, and I expect to report further along these lines following our WESCON meeting.

On May 11th, I attended a meeting of the Professional Groups Committee (composed of chairmen of all the Professional Groups, 14 members-at-large, and the executive committee) in New York. A principal subject for discussion was an application for admission to Group status of a proposed Professional Group on Atomic and Molecular Engineering. Dr. George Sziklai and Dr. Allen Matthews made a thoughtful and well-planned presentation favoring group status. However, emphasizing a more recently enforced policy of strictness in authorizing the admission of new groups, the application was denied on vote of the committee membership with a recommendation from Dr. Baker that this particular activity continue to be nourished under the auspices of the PGED. I believe that it was generally felt that a separate group would eventually evolve but that it might now be too early to require it to function alone.

Under New Business, considerable discussion was given to the subject of the quality of papers at the national conventions, with substantial support being given to the feeling that there was little prestige to be gained in reading a paper at either the annual IRE convention or at the WESCON meeting. A number of means were discussed toward improvement of the quality of papers, including restriction of the number of papers, more rigorous selection and screening, and a suggestion by the Professional Group on Engineering Writing and Speech that manuscripts be reviewed by that group prior to presentation. The problem of the quality of papers is not peculiar to the "big" conventions only, but to one degree or another affects any technical convention. Although PGEC usually enjoys a high standard of quality from its membership at the Joint Computer Conferences, the problem is ours too, and particular attention will be addressed to it by the administrative committee in the future. Your thoughts on the subject are certainly encouraged.

Finally, I want to extend a welcome to our newest Chapter, within the Omaha-Lincoln Section.

Richard O. Endres
Chairman 1959–1960

ONR Symposium on Microwave Techniques for Computing Systems*

The Editor gratefully acknowledges the cooperation and assistance of R. L. Wigington, Program Chairman for the ONR Symposium, and of Miss Betty Jo Ellis, Vice-Chairman of the Washington (D. C.) Chapter of PGEC, in making these papers available for prompt publication in these TRANSACTIONS as a unified set. Our thanks also to Marshall Yovits for preparing a suitable background note to introduce the papers.

The reader's attention is called to the paper immediately following the ONR Microwave Symposium papers in this issue, "The Parametron Digital Computer MUSASINO-1," by S. Muroga and K. Takashima, which describes a computer using low-frequency techniques analogous to the microwave techniques discussed in some of the ONR papers.

—The Editor

On March 12, 1959, the Information Systems Branch of the Office of Naval Research, recognizing the great interest that has recently evolved in this country and abroad in the applications of microwaves to the field of computers and associated systems, sponsored a Symposium on Microwave Techniques for Computing Systems in Washington, D. C. The symposium was attended by about 400 people from industry, government, and universities, and appeared to be well received.

The papers at this meeting were entirely invited and were felt by the committee to be representative of the efforts now in progress in this country. It was felt to be desirable (perhaps incorrectly) to limit the symposium to one day, and, accordingly, a number of investigators who perhaps should have been were not invited to contribute. Certain associated topics were not considered at all, although in a longer meeting it would have been most useful to do so. For example, bulk storage techniques were considered only in a very cursory fashion, although techniques for storing large numbers of microwave pulses are presently under study. To those who were not included in the program, the committee offers its apologies.

The sponsorship of the work leading toward a computer operating at microwave frequencies is diverse. Some of the work, particularly in the peripheral areas, has been industrially sponsored, much of the work has been sponsored by the Navy Bureau of Ships, and a smaller amount has been sponsored by the Office of Naval Research. One of the earliest efforts in this field, which began in June, 1956, was sponsored by the ONR at the General Electric Microwave Laboratory in Palo Alto, Calif.

The committee responsible for the program and arrangements of the symposium was made up of M. C. Yovits, Chairman, and G. D. Goldstein, both from the Office of Naval Research, and R. L. Wigington and Lieutenant Paul Peters of the Department of Defense.

In addition to the papers presented in full in this issue of the IRE TRANSACTIONS ON ELECTRONIC COMPUTERS, an abstract is included of a paper prepared by Norman Kroll and presented by Sol Krongelb, both of the International Business Machines Corporation, entitled "Properties of Propagating Structures with Variable Parameter Elements." This paper will be presented elsewhere for publication.

MARSHALL C. YOVITS
Symposium Chairman
Information Systems Branch
Office of Naval Research
Washington, D. C.

* Manuscript received by the PGEC, April 30, 1959.

History and Introduction—Microwave Techniques for Computers*

R. E. MEAGHER†

IN our quest for faster computers, it seems natural and appropriate to consider microwave techniques. Because of the very high frequencies and especially the wide bandwidths available in these techniques, they are enticing to the computer engineer as a possible solution to the need for faster computing systems. However, there are serious problems in the direct application of the well-developed radar and communication microwave techniques to computers. For example, it would not be very practical to make the 10,000 or so interconnections of a conventional general-purpose computer with the standard waveguide—even in the relatively small K band size! In order to consider some of these questions and to provide a background for the more detailed technical papers which are to follow, we will review our needs very briefly and, in very broad terms, discuss some of the work which has been carried out.

Let us first examine the need for new techniques which has resulted from our desire for higher speed. The existing circuits, with separate resistors, diodes, capacitors, and transistors, have a physical size which requires at least one cubic inch for a logical element. One such circuit within itself has a loop which constitutes an inductance with a shunt capacitance in the switching element. This LC circuit exhibits resonance. If the loop is about one-half inch in diameter, the inductance would be about 0.06 μh, and further, if the capacitance is 5 $\mu\mu$fd (both reasonable minimum values), the resulting resonant frequency would be about 300 mc. Clearly, it would be difficult to operate this circuit at an information frequency of more than about one-fifth the resonant frequency, or in other words, 60 mc. We are already close to this frequency in some present computer circuits. Thus, faster circuits require either smaller size for "lumped-constant" techniques or, alternatively, "distributed-constant" techniques. Low-temperature circuit elements offer one possibility for extremely small size. Microwave techniques, the subject for this Symposium, offer the possibility of the distributed-constant approach.

It appears that the earliest serious consideration of the application of microwave techniques to computers was that made by Dr. Edson at Stanford University. He recognized that multimode oscillators might be used as memory devices [16]. This work was, in effect, continued at the General Electric Microwave Laboratory and the Stanford Research Institute with the idea of using a separate carrier frequency for each of several bits in a single memory or computing channel [18]–[20]. Specifically, one microwave channel could handle one frequency for each of the 10 decimal digits in a single channel, and methods of adding and storage in a circulating system were suggested. All of these studies may be thought of as having used more or less conventional microwave components, but with number representations and computer logic to make use of the wide bandwidths and amplitude modulation.

There has been work done at Bell Telephone Laboratories [3], [8], [12], at Stanford University [7], [17], and at the Diamond Ordnance Fuze Laboratories [10], [11] on the problems of microwave pulse generation and amplification. These efforts were less directed to computer circuitry. However, the combination of these pulse techniques and the number representation, logic and memory techniques of Stanford University and the General Electric Microwave Laboratory represents a straightforward but important approach to the use of conventional microwave components for computers.

The late Professor John von Neumann had a new and different idea on how to get high switching speeds using microwave frequencies which were relatively simple devices compared to conventional microwave amplifying devices. This new idea is recorded in his patent (assigned to IBM) which was filed in April, 1954 and issued in December, 1957 [1]. His suggestion was to use nonlinear elements in resonant circuits to make possible the coupling of energy from one frequency as a source to a second lower frequency which represented the information. Professor von Neumann's computer circuit may properly be called a phase-locked subharmonic oscillator. It is interesting to note that he suggested the nonlinear capacitance of a semiconductor junction as the nonlinear element in the circuit because it could easily be small and used at extremely high frequencies. Many earlier examples can be found in which the power at one frequency is converted to another in both mechanical and electrical phenomena. Subharmonic and higher harmonic generators depend upon this effect, and only a perfectly linear circuit would be without some possible harmonic generation.

The idea of a phase-locked subharmonic oscillator for computers arose independently in Japan from Dr. E. Goto's work with Professor H. Takahashi in the

* Manuscript received by the PGEC, April 30, 1959. Presented at the ONR Symp. on Microwave Techniques, Washington, D. C., March 12, 1959. This work has been supported in part by the ONR under Contract Nonr-1834(15).

† Digital Computer Lab., University of Illinois, Urbana, Ill.

Department of Physics at Tokyo University [2], [4], [9]. In his paper, Dr. Goto indicates that under certain conditions this circuit is described by Mathieu's equation [15], which arose from a study of vibrations of an elliptic membrane and was reported on in 1868. Dr. Goto named the computer element he proposed the "parametron." This device has been used by Dr. Saburo Muroga to make a general-purpose digital computer for the Nippon Telegraph and Telephone Public Corporation [14]. Now in operation for two years, this computing machine has an order code which is substantially the same as the Illiac at the University of Illinois, although the electrical circuits use 5356 parametrons in place of an approximately equal number of triodes. These parametrons do not operate at microwave frequencies and use nonlinear inductances rather than the nonlinear capacitances which now definitely appear to be favored for microwave techniques.

Dr. Muroga has suggested a mechanical analogy which helps to explain how these phase-locked subharmonic oscillators, or parametrons, work. A child can swing on a swing and "pump" himself to make up for the losses which occur. He does this by raising his center of gravity higher than it would normally be at the ends of the excursion for a simple pendulum. Thus, he increases the potential energy at the end of each swing (and the motion is no longer "simple harmonic"). For each period of the swing, the child may raise his body twice; if, therefore, the fundamental frequency of the system for small amplitudes is f, the pumping frequency is $2f$. Furthermore, it should be noted that for a given pumping motion at frequency $2f$, the swing can be in either of two phase states which are 180° apart. The actual phase state in which the swing finally oscillates depends upon some extremely small disturbance in the very beginning. This little disturbance, in one phase or the other at frequency f, is amplified by pumping at frequency $2f$. Oscillation in one of these phase states may be used to represent "0" in the binary number system and oscillation in the other phase state may be used to represent "1" in the binary number system. In Dr. Muroga's computer, the Musasino I, the frequency f is 1.2 mc and the pumping frequency is 2.4 mc. Since the power source and information frequencies differ by a factor of two, it is not too difficult electrically to distinguish them and to keep them separate.

Anyone who does not have a swing may demonstrate this same phenomenon by making a simple pendulum with a string and a bob. If the pendulum string is now supported through a small hole, one may pull horizontally on the string at a frequency twice that of the pendulum and after a slight disturbance achieve a large pendulum amplitude without any motion of the supporting hole. It's fun; try it![1]

Recently, at the IRE-AIEE University of Pennsylvania Solid-State Circuits Conference, F. Sterzer and W. R. Beam of RCA Laboratories described a phase-locked subharmonic oscillator in which f was 2000 mc and $2f$ was then 4000 mc—very acceptable microwave frequencies.

It is clear that the phase-state representations for zero and one may be combined to form AND and OR circuits, whose signal could excite a succeeding phase-locked oscillator. By inversion of a signal, the NOT can be formed. Each phase-locked oscillator is really called a flip-flop in computer terminology. Ordinarily, the driving power at frequency $2f$ would need to be removed for a period of time before such a device could change state. Similarly, the power must be applied for a similar period of time in order to have the oscillator assume the proper amplified amplitude of any new state. It is important that these two statements are understood because behind them is the limitation on the speed with which the information may be processed. It would be reasonable to say that a minimum of 10 cycles of the fundamental frequency would be required to change from one state to another. This would suggest an information frequency of $f/10$ or 200 mc for the phase-locked oscillator described by Sterzer and Beam. In Muroga's computer, the information frequency is 10 kc, or 120 times less than the fundamental frequency f. This may be a more realistic ratio between fundamental and information frequencies than the number 10. This ratio and its governing factors are especially important characteristics of any new parametrons.

The question of size for these new elements is still important. If the fundamental frequency could be made 10,000 mc, then the information frequency would be about 1000 mc, and we would have one binary operation per millimicrosecond. In a general-purpose computer where many elements of this speed need to be interconnected, the free-space propagation time of about 1 mμsec per foot will need to be considered. If an OR circuit or a flip-flop were one foot from the place or places to which information is to be sent, the over-all time to complete an operation would be 2 mμsec. Thus, under ordinary circumstances, the machine would be slower by a factor of two if the propagation time between elements were equal to the operation time per element. This suggests that for computing machines with many interconnections of elements per unit, as is commonly the case, devices which process information at 1000 mc should be small enough to contain hundreds or thousands of them in one cubic foot. Relatively simple computing groups might be able to allow for all the propagation delays while still keeping the information rate high, but this would appear to be more acceptable for special-purpose machines with a modest number of switching elements. In general, a high premium will be placed on small size.

New semiconductor diodes appear to have advantageous characteristics for phase-locked oscillator use. Together with the use of layer or "strip-line" types of construction, they show hope for a much smaller physical size with microwave techniques for computers than has been achieved in the past for radar or communications.

[1] The author first saw this demonstrated by Dr. J. Rajchman of RCA Laboratories.

References

Publications

[1] J. von Neumann, "Nonlinear capacitance or inductance switching, amplifying, and memory organs," U. S. Patent 2,815,488; December 3, 1947.
[2] E. Goto, "New parametron circuit element using nonlinear reactance," *KDD Kenkyu Shiryo*, no. 52; November, 1954. (In Japanese.)
[3] C. C. Cutler, "The regenerative pulse generator," PROC. IRE, vol. 43, pp. 140–148; February, 1955.
[4] E. Goto, "On the application of parametrically excited nonlinear resonators," *J. Inst. Elec. Commun. Engrs. (Japan)*, vol. 38, pp. 770–775; October, 1955. (In Japanese.)
[5] S. Oshima, "Introduction to parametron," *Kenshi Kogyu*, vol. 4, no. 11, p. 4; 1955. (In Japanese.)
[6] S. Ohima, Enomoto, and Watanabe, "Oscillation theory of parametron and method of measuring nonlinear elements," *KDD Kenkyu Shiryo*, no. 90; November, 1955.
[7] F. S. Coale, "A switch detector circuit," IRE TRANS. ON MICROWAVE THEORY AND TECHNIQUES, vol. MTT-3, pp. 59–61; December, 1955.
[8] O. E. DeLange, "Experiments on the regeneration of binary microwave pulses," *Bell Sys. Tech. J.*, vol. 35, pp. 67–90; January, 1956.
[9] H. Takahaski, "The parametron," *Tsugakkat Shi*, vol. 39, no. 6, p. 56; 1956. (In Japanese.)
[10] M. A. Armistead, E. G. Spencer, and R. D. Hatcher, "Microwave semiconductor switch," PROC. IRE, vol. 44, p. 1875; December, 1956.
[11] R. V. Garver, E. G. Spencer, and R. C. LeCraw, "High speed microwave switching of semiconductors," *J. Appl. Phys.*, vol. 28, pp. 1336–1338; November, 1957.
[12] C. A. Burrus, "Millimicrosecond pulses in the millimeter wave region," *Rev. Sci. Instr.*, vol. 28, pp. 1062–1065; December, 1957.
[13] Z. Kiyasu, "Parametron," *J. Inst. Elec. Commun. Engrs. (Japan)*, vol. 41, pp. 397–403; April, 1958. (In Japanese.)
[14] S. Muroga, "Elementary principle of parametron and its application to digital computers," *Datamation*, vol. 4, pp. 31–34; September/October, 1958. Also S. Muroga and K. Takashima, "The parametron digital computer MUSASINO-1," this issue, p. 308.
[15] E. T. Whittaker and G. N. Watson, "A Course of Modern Analysis," Cambridge University Press, Cambridge, Eng., pp. 404–428; 1935. N. W. McLachlan, "Theory and Application of Mathieu Functions," Oxford University Press, New York, N. Y.; 1947. N. W. McLachlan, "Ordinary Nonlinear Differential Equations," Oxford University Press, New York, N. Y.; 1950.

Reports

[16] W. A. Edson, "Frequency Memory in Multi-Mode Oscillators," Stanford University, Stanford, Calif., Tech. Rept. No. 16; July 19, 1954.
[17] D. J. Grace, "A Microwave Switch Employing Germanium Diodes," Applied Electronics Lab., Stanford University, Stanford, Calif., Tech. Rept. No. 26; January, 1955.
[18] M. P. Forrer, "Investigation of Application of Frequency Memory Techniques," General Electric Microwave Lab., Palo Alto, Calif., Rept. No. R55 ELS 12.3; December 12, 1955.
[19] R. H. Baker, T. H. Meisling, and G. Nielson, "Application of Frequency-Domain Techniques to Computers," Stanford Res. Inst., Menlo Park, Calif.; September, 1956.
[20] M. P. Forrer, L. Fein, and V. Met, "Microwave Techniques for Computer Application," General Electric Microwave Lab., Palo Alto, Calif., Quart. and Final Repts. on ONR Contract Nonr-2127(00); December, 1956–December, 1958.

Papers

[21] F. Sterzer and W. R. Beam, "Parametric Sub-Harmonic Oscillators," IRE-AIEE-University of Pennsylvania Solid-State Circuits Conf., Philadelphia, Pa.; February 13, 1959.

Nanosecond* Logic by Amplitude Modulation at *X* Band**

W. C. G. ORTEL†

Summary—**A basic circuit, consisting of a diode modulator controlled by the signal from a diode detector, may perform logical AND, EXCLUSIVE-OR and OR functions upon pulsed microwave signals. Pulse rates up to 500 mc have been used at a carrier frequency of 11,000 mc. To demonstrate that microwave circuits may be used for the regeneration and circulating storage of pulses, as well as for logic, a digital arithmetic unit has been built which multiplies two 8-digit binary numbers. Various forms of the basic circuit have been studied in operation.**

Introduction

A serial digital computing system has been studied, in which information is transmitted in the form of a pulsed microwave signal. Signals of this form may be conveniently combined in logic circuits which consist of detectors and modulators. In order to demonstrate this amplitude-modulation computing scheme, a serial accumulator has been built which performs binary multiplication.

The basic logical element used is shown in Fig. 1. Signal C is the RF input to a modulator having two control inputs derived by detecting RF signals A and B. The binary digits, one and zero, are represented by the presence or absence of an input pulse of standard amplitude and duration. If the modulator is of the balanced type, RF power from C will be transmitted to the output if just one (but not both) of the signals A and B are present. If C is always present, the output is the EXCLUSIVE-OR function of A and B. If B is never present, the output is the AND function of A and C. If B and C are always present, the output is the NOT function of A. If the modulator is not of the balanced type, the output is the OR function of A and B.

The particular form of modulator used is a bridge circuit using a waveguide hydrid junction. Signal C is applied to terminal 1 of the junction, and the output is taken from the conjugate terminal 3. Side arm 2 is terminated in a semiconductor diode, M_a, whose admittance may assume one of two values Y_a or Y_a', depending on its bias. Side arm 4 contains a similar diode,

* Although the term *millimicrosecond* has attained some currency, the metric system provides the standard prefix *nano-*, (abbreviated n-), by which the decimal multiplifier 10^{-9} may be denoted conveniently without multiple prefixes.

** Manuscript received by the PGEC, April 20, 1959. Presented at the ONR Symp. on Microwave Techniques for Computing Systems, Washington, D. C., March 12, 1959.

† Bell Telephone Laboratories, Inc., Murray Hill, N. J.

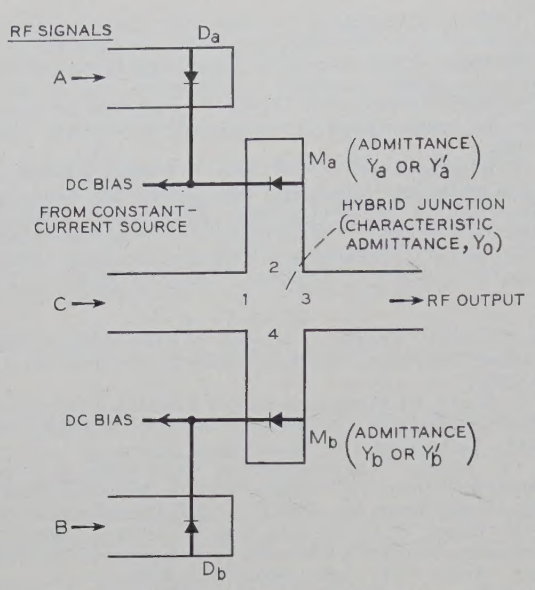


Fig. 1—Basic logic circuit. Modulating diodes M_a and M_b are isolated from detectors D_a and D_b at microwave frequencies.

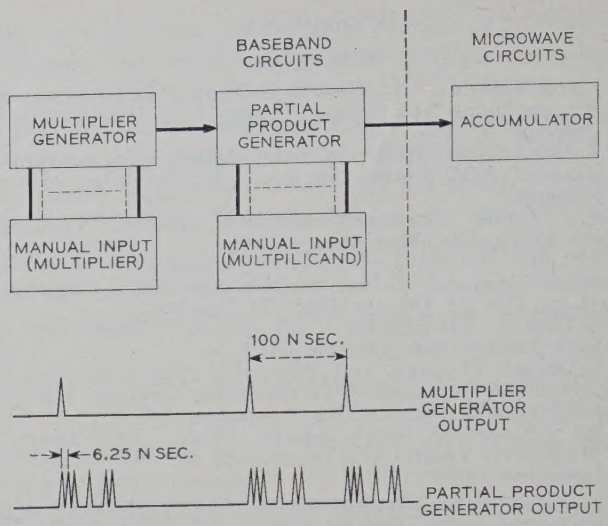


Fig. 2—Digital arithmetic unit arranged to perform multiplication. The baseband circuits generate a series of partial products which are added in the microwave accumulator.

M_b, whose admittance may be Y_b or Y_b'. If $Y_a = Y_b$ and $Y_a' = Y_b'$, the modulator is of the balanced type necessary to produce the EXCLUSIVE-OR function. If $Y_a' = Y_b'$, but $Y_a \neq Y_b$, the modulator is not of the balanced type, and the OR function may be produced.

Bias for diodes M_a and M_b is derived from signals A and B, rectified in other diodes D_a and D_b. When signal A is present, diode M_a has admittance Y_a. When A is absent, M_a has admittance Y_a'. Similarly, M_b has admittance $Y_b(Y_b')$ when signal B is present (absent).

If operated at the proper power level, the circuit discriminates against small input signals and limits large ones. Furthermore, when input C is not required to contain information, it may be modulated by clock pulses. Thus, if the input pulses are slightly degraded, the output pulses may be partly restored to standard amplitude and time.

When an X-band (11,000-mc) carrier is used, the bandwidth of the microwave circuits is sufficient to handle pulse rates as high as 1000 mc. Furthermore, the time constant of the coupling network between detector and modulator is compatible with such pulse rates. Since, however, the output power of the modulator is about 20-db less than the input to a detector, each logic circuit must be followed by an amplifier. The traveling-wave amplifiers which have been used introduce a delay of about 10 nsec: thus, at pulse rates above 100 mc, it is necessary to devise a logical design under the restriction that the shortest interstage delay is greater than one pulse interval.

In order to explore the problems of this computing system, a machine has been built which generates serially the 16-digit product of two 8-digit binary numbers. Microwave circuits are used in the accumulator portion of the machine. In the accumulator, interstage delays of several pulse intervals are tolerated by a novel logical design. The multiplication process occupies more digit intervals with this design than it would in a standard serial accumulator. Nevertheless, the higher digit rate which is permitted with the present accumulator more than compensates for this disadvantage, and there is a net increase in the speed of multiplication.

The machine performs binary multiplication according to the following scheme. The 16 digits of the multiplicand and multiplier are required as inputs. These are inserted in parallel by setting 16 manual switches (see Fig. 2). In one multiplication cycle, the multiplier is converted to serial form, the least significant digit first, in the multiplier generator. Its output has an accurately timed digit spacing of 100 nsec and a pulse width of 5 nsec. The multiplier pulses enter the partial product generator, which has as control inputs the 8 digits of the multiplicand. Each time a multiplier pulse is applied, the partial product generator produces the digits of the multiplicand serially, the least significant digit last, with a pulse spacing of 6.25 nsec. The spacing of the multiplier pulses insures that the successive appearances of the multiplicand are in a suitable time relationship for use as partial products. These partial products are inserted into an accumulator of the circulatory type, which forms and stores their sum. Since successive partial products must be shifted by one digit interval before addition, the circulating time of the accumulator is 106.25 nsec, 6.25 nsec greater than the interval between partial products.

The shortest interstage delay in the accumulator is 50 nsec. A standard serial accumulator with this minimum delay would be limited to a pulse rate of 20 mc, and would require 6.4 μsec to complete the multiplication of two 8-digit numbers. Through a novel logical design, the microwave accumulator requires 1.6 μsec for this process, operating at a pulse rate of 160 mc. It would be feasible, with the present microwave circuits, to increase the pulse rate to 640 mc, which would reduce the multiplication time to 0.4 μsec.

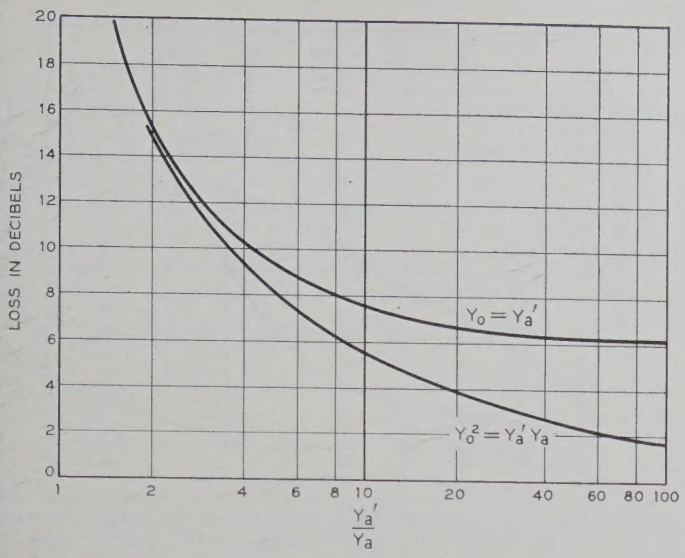


Fig. 3—Power loss between input C of the modulator of Fig. 1 and the output. When M_a and M_b each have admittance Y_a', the loss is infinite. When M_a is switched to admittance Y_a, the loss is a function of Y_a'/Y_a and of Y_0, as shown. The two choices of Y_0 correspond to equations (2) and (3).

Basic Logic Circuit

Since the modulator is a basic part of the general logic circuit, we shall consider its operation in more detail. Suppose that RF power P is incident at terminal 1 of the hybrid junction. The output voltage at terminal 3 is simply the vector difference of the reflected waves in arms 2 and 4, which are terminated in diodes M_a and M_b. If M_a has admittance Y_a', and M_b has admittance Y_b', the reflected waves cancel and there is no output, provided $Y_a' = Y_b'$. If M_a has admittance Y_a, and M_b has admittance Y_b', and if the characteristic admittance of the transmission line is Y_0, the output power in arm 3 is

$$P_0 = \frac{1}{4} P \left| \frac{Y_0 - Y_a}{Y_0 + Y_a} - \frac{Y_0 - Y_b'}{Y_0 + Y_b'} \right|^2 . \qquad (1)$$

(Y_a' and Y_b' correspond to forward bias states; Y_a and Y_b to back-bias.) In practice, the diode holders are provided with reactive elements (a short-circuiting plunger and screws). While it is, perhaps, most natural to think of these elements as modifying the diode admittances, an equivalent point of view is to consider the diode admittances fixed, and to think of the elements as modifying Y_0. A natural choice of Y_0 in (1) would be $Y_0 = Y_a' = Y_b'$, so that the diodes are matched to the transmission line in the forward-bias state. In this case, the output power is

$$P_0 = \frac{1}{4} P \left[\frac{Y_a'/Y_a - 1}{Y_a'/Y_a + 1} \right]^2 . \qquad (2)$$

With this arrangement, the maximum value of P_0/P (for large values of Y_a'/Y_a) is $\frac{1}{4}$. Examination of (1) shows that the best choice of Y_0 is $Y_0 = \sqrt{Y_a' Y_a}$, in which case the output power is

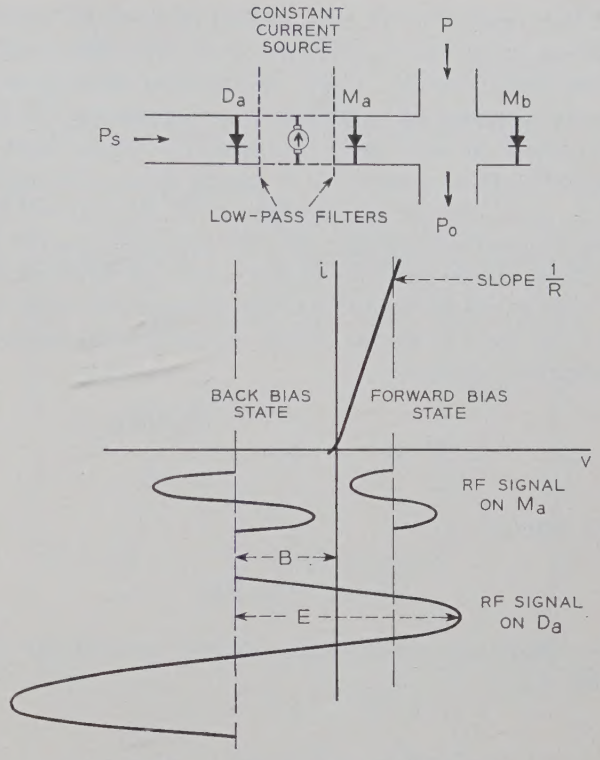


Fig. 4—Operation of combined detector and modulator circuits. When P_s is absent, D_a and M_a are forward-biased by the constant-current source, and the RF admittance of M_a is high. When P_s is present, the rectified current through D_a establishes a back-bias voltage B on D_a and M_a, and the RF admittance of M_a is low.

$$P_0 = P \left[\frac{\sqrt{Y_a'/Y_a} - 1}{\sqrt{Y_a'/Y_a} + 1} \right]^2 . \qquad (3)$$

Here, for large values of Y_a'/Y_a, $P_0/P = 1$. Eqs. (2) and (3) are plotted in Fig. 3 for comparison.

We shall now proceed to consider the combined detector and modulator circuits. We assume that the detector and modulator diodes, D_a and M_a, are identical, and that they have the simple current-voltage relationship shown in Fig. 4; a constant forward resistance, R, for all values of forward bias, and an infinite back resistance. Bias is supplied to both D_a and M_a from an external source with a constant-current characteristic. In the absence of RF signals, current from the external source divides equally between D_a and M_a, maintaining both diodes in the forward-bias state. When an RF signal is incident on D_a, conduction during the peak of the cycle provides a dc current to back-bias it during the remainder of the cycle. Since M_a is now back-biased, the rectified current through D_a is equal to the current from the external source. The values of forward and back bias established on M_a by the absence or presence of an RF signal on D_a must be great enough so that any RF signal on M_a does not cross the zero-bias point during any part of the RF cycle. Thus M_a never acts as a rectifier. It is evident that we require the RF signals of D_a to be greater than the signals on M_a.

For simplicity, we consider the modulator to be adjusted so that $Y_a' = Y_0$ ($Y_a' = 1/R$). The forward-bias

current supplied to M_a must exceed the peak RF current through it, which is $\sqrt{P/R}$, while the back-bias voltage must exceed $2\sqrt{PR}$. Thus, the detector is required to supply a current of $2\sqrt{P/R}$ at a voltage of $2\sqrt{PR}$. It is of interest to find the RF power, P_s, which must be applied to D_a as a signal. If a voltage $E \cos \omega t$ is applied to D_a, and if the back-bias voltage is B, current will flow only during that portion of the RF cycle where $\omega < \epsilon$; cos $\epsilon = B/E$. The dc component of current will be $E/\pi R$ (sin $\epsilon - \epsilon \cos \epsilon$), while the admittance at frequency $\omega/2\pi$ will be $1/\pi R$ ($\epsilon - 1/2$ sin 2ϵ). Thus, we require the detector to supply a current

$$E/\pi R \, (\sin \epsilon - \epsilon \cos \epsilon) = \frac{2\sqrt{2PR}}{R}$$

at a voltage

$$E \cos \epsilon = 2\sqrt{PR}$$

from which we obtain tan $\epsilon - \epsilon = \pi$ or $\epsilon = 1.35$. At this value of ϵ,

$$1/\pi R(\epsilon - 1/2 \sin 2\epsilon) = \frac{0.362}{R} \cdot$$

If the detector is adjusted for a match,

$$E = \sqrt{\frac{2P_sR}{0.362}} \quad \text{and} \quad \frac{P_s}{P_0} = 60.5.$$

This simple analysis leads to a value of 18 db for the loss between signal input and output of a logic circuit. A more complicated model of the diode leads to a similar result. In practical circuits, a loss of 20 db is typical. It is possible to device circuits with different biasing arrangements, and to make the forward resistance of the detector different from that of the modulator; however, neither theory nor experiment has disclosed any such arrangement which is substantially better than that described.

The input-output characteristics of a detector-modulator combination were measured in some detail. Diodes of type 1N78 were used as M_a and D_a of Fig. 4. The diode holders each included a low-pass filter; the total capacitance of the connecting link between diodes was 6 pF. A total forward-bias current of 2 ma was supplied from a battery in series with a 1000-ohm resistor. The diode holders and hybrid junction were composed of RG-52/U waveguide. For measurement purposes, diode M_b was replaced by a precision attenuator and a calibrated short-circuiting plunger, which served as a reference admittance of known magnitude.

First, with the reference admittance set equal to Y_a', the output power was measured as a function of CW input power at A and C. Typical results are shown in Figs. 5 and 6. There is a pronounced minimum in the curves of Fig. 6, since the forward bias applied to the

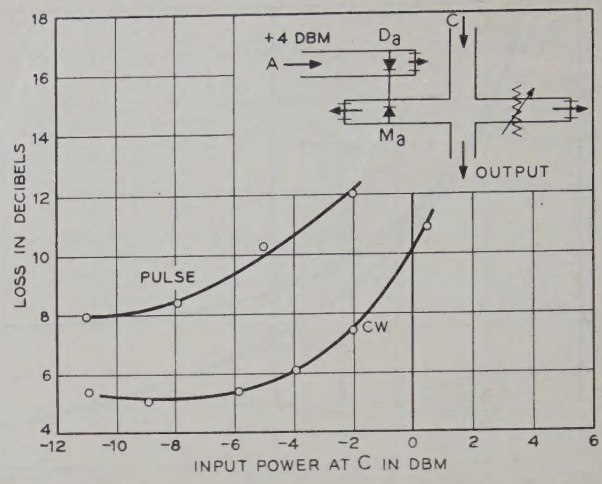


Fig. 5—Loss from input C of logic circuit to output. Input A is fixed at +4 dbm. Diode M_b of Fig. 1 is replaced by a passive load.

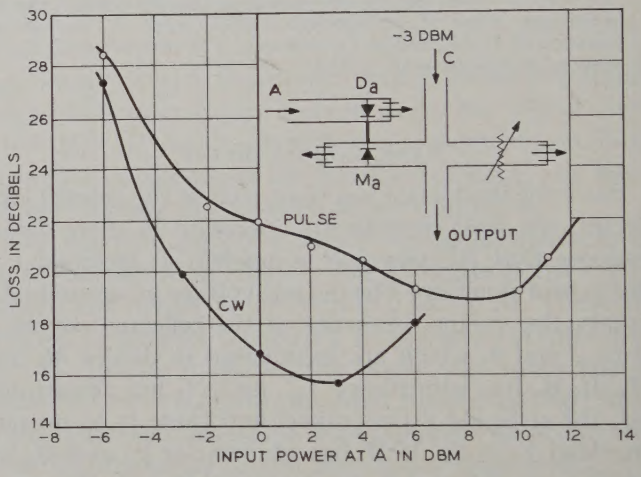


Fig. 6—Loss from input A of logic circuit to output. Input C is fixed at −3 dbm.

detector discriminates against small input signals at A, while the output reaches a limiting value when the reverse bias of the modulator diode equals the peak RF voltage.

Second, the admittance of M_a was measured under various CW input conditions by observing the setting of the reference admittance for zero output power. Fig. 7 shows this result.

The above measurements were then repeated with pulsed inputs. Identical pulses (9 nsec wide at the base, at 100-nsec intervals) were applied to the logic circuit and to a precision attenuator. The output paths, of different lengths, were combined, allowing the two output pulses to be seen on the same oscilloscope sweep. The push-pull deflection system of the traveling-wave oscilloscope was driven by two 1N78 diode detectors of opposite polarity, eliminating the need for a pulse-inverting transformer. The detection apparatus was calibrated with respect to the saturation point of a traveling-wave amplifier, on the assumption that this point is the same for CW and pulsed operation. Loss measurements made in this fashion are also shown in Figs. 5 and 6. Admit-

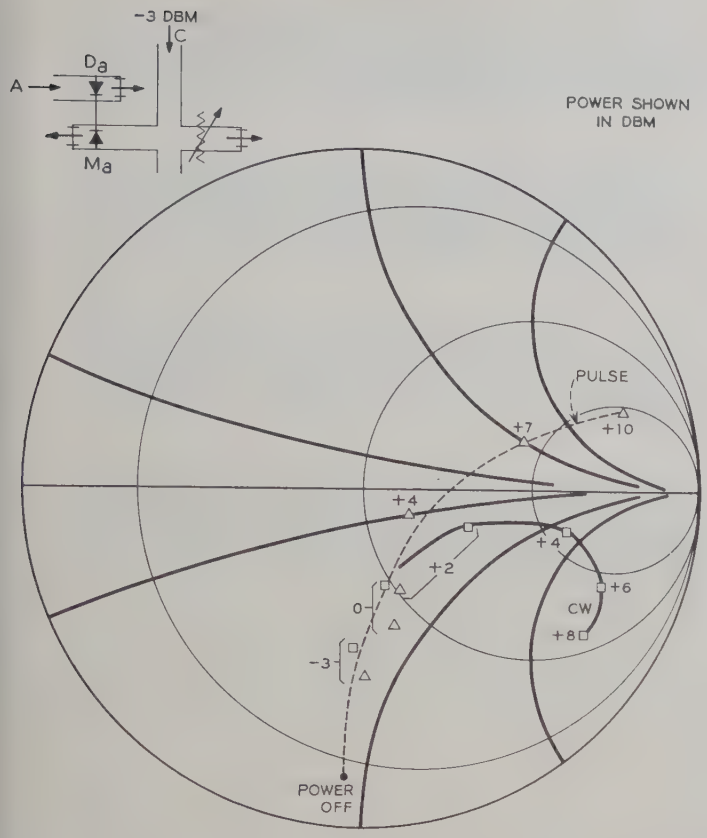


Fig. 7—Admittance of M_a, measured with respect to an arbitrary reference plane. The corresponding value of input power at A is given at each point. Input C is fixed at −3 dbm.

tance measurements were made by adjusting the reference admittance until an inverted pulse of maximum amplitude was observed. They are shown in Fig. 7.

The loss measurements show that if the output of such a circuit is to be amplified and applied to the input of a similar circuit, the interstage gain should be about 22 db. Small noise pulses will then suffer a net loss (provided that they are smaller than 5 dbm), while signal pulses (greater than 5 dbm) will be partially restored to a standard amplitude of 10 dbm.

Microwave Accumulator

The adder in the dynamic accumulator consists, as usual, of two half-adders, interconnected through an OR circuit; however, the delays in the interconnecting links are arranged in an unconventional manner. In a usual full serial adder, the inputs enter with the least significant digit first, and a given sum digit is produced immediately when the corresponding input digit appears. A carry signal derived at that time from one or the other half-adder is delayed by one digit interval, τ, so as to re-enter together with the next (more significant) input digit. Since the two outputs of a half-adder are mutually exclusive, two carries cannot be generated simultaneously.

To form a standard accumulator, the adder output would be delayed and re-entered as one of its two inputs, as in Fig. 8. The accumulator cycle time could be

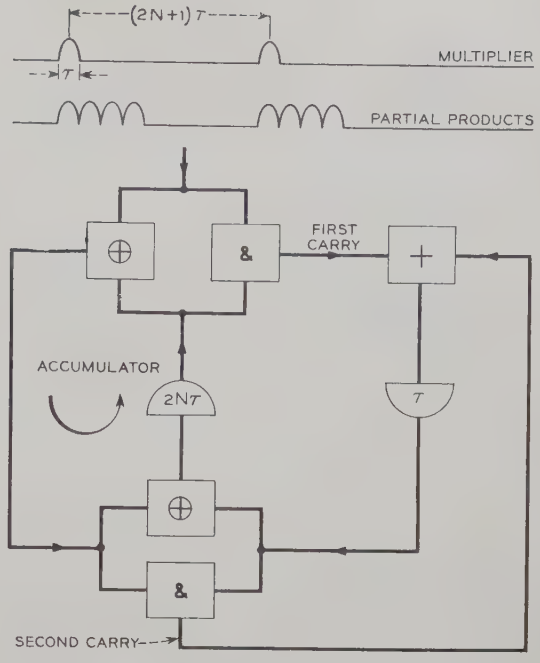


Fig. 8—Standard serial accumulator, with inputs properly timed for multiplication.

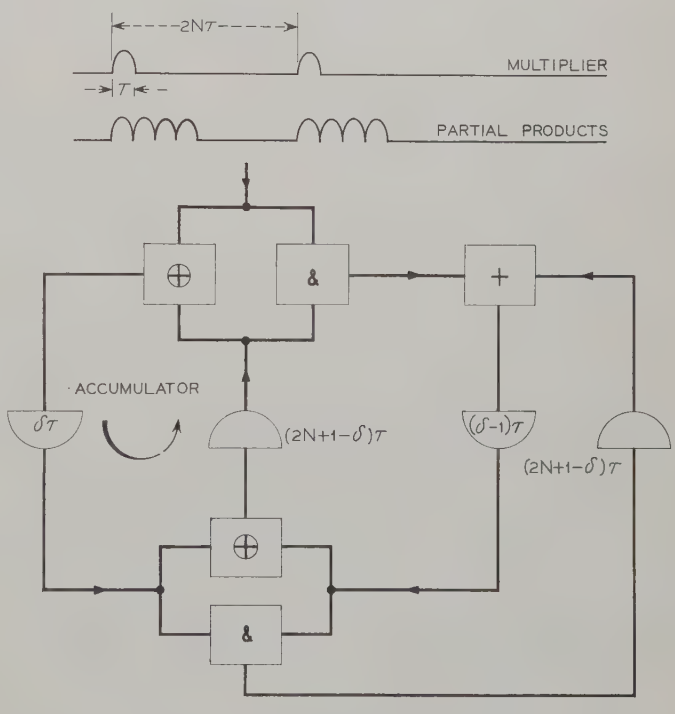


Fig. 9—Serial accumulator, modified to make allowance for large interstage delays.

as short as $2N\tau$, for N-digit inputs. In this case, the multiplier pulse interval would have to be $(2N+1)\tau$.

When a delay greater than τ is associated with each logic circuit, the proper timing of carries may be maintained by the arrangement of delays shown in Fig. 9. An arbitrary delay, $\delta\tau$, is introduced between EXCLUSIVE-OR circuits, and compensating changes made in the accumulator loop and in the "first carry" path. Now a "second carry" is not available soon enough to be re-

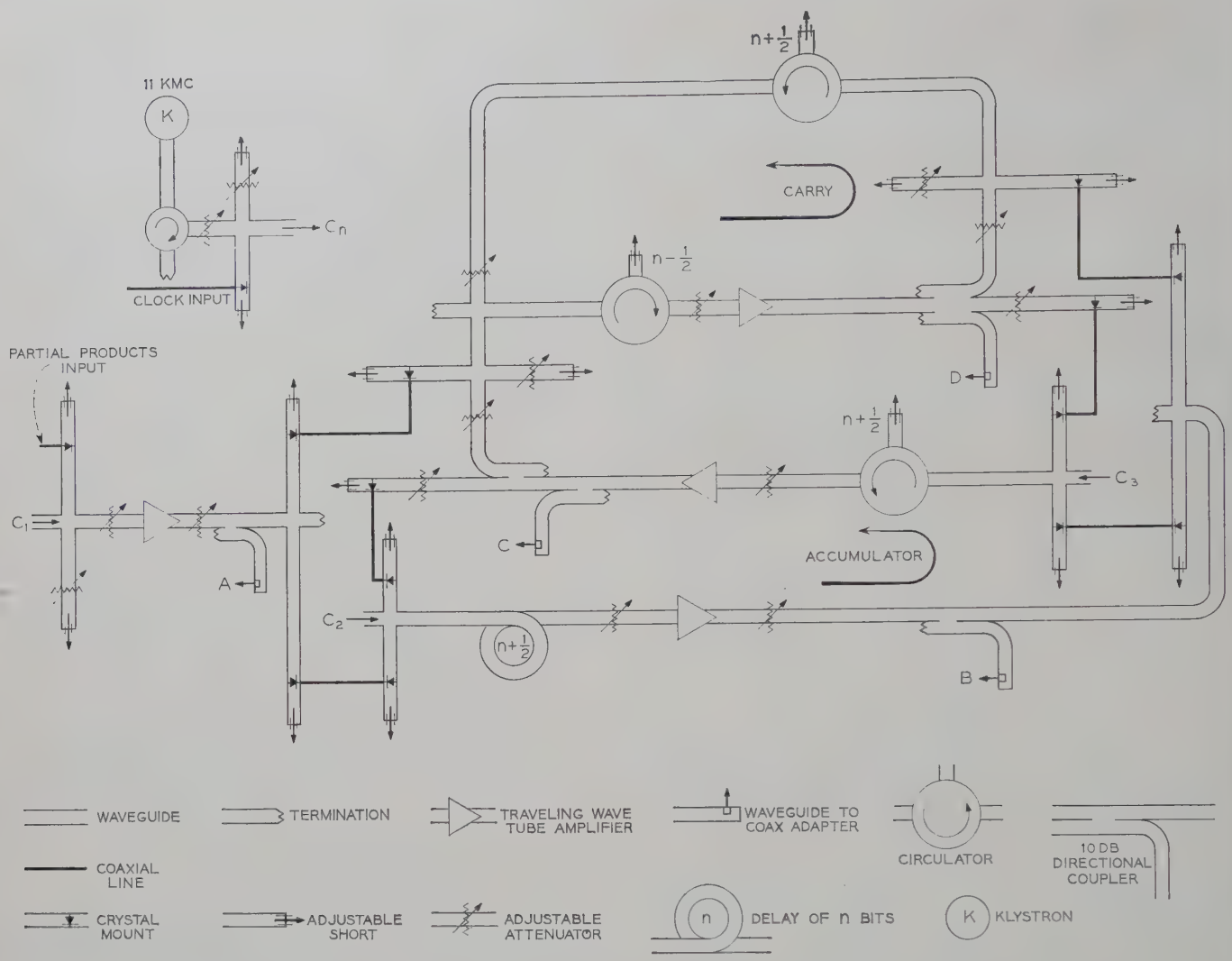


Fig. 10—Microwave circuits of the accumulator. The delay arrangement of Fig. 9 is used, with $\delta = N + \frac{1}{2}$.

inserted in the same accumulator cycle in which it is generated, so it is delayed until the proper time of the next accumulator cycle. It may require N accumulator cycles before the result of one addition process is present in the accumulator; however, it is not necessary to wait for the completion of one addition process before starting another. The two outputs of the lower half-adder are mutually exclusive, so first and second carry pulses will never reach the OR circuit simultaneously.

With the delays as shown in Fig. 9, the least significant digit of the multiplicand must come *last;* each second carry must be advanced to an earlier time in the next accumulator cycle.

The total time required to multiply two N-digit numbers with the standard accumulator of Fig. 8 is approximately $2N^2\tau$, while with that of Fig. 9, it is about $4N^2\tau$. With the latter scheme, however, a higher digit rate, $1/\tau$, is permitted for a given value of interstage delay, since the shortest interstage delay is $N\tau$ rather than τ. Thus, if the only limitation on speed is the interstage delay, the multiplication time is reduced by the factor $(1/2)N$ through the novel delay arrangement of Fig. 9.

The output of the partial product generator of Fig. 2 is applied to a diode modulator, similar to that in the microwave logic circuits, to produce RF pulses suitable for use as an input to the accumulator.

Fig. 10 shows the microwave circuits of the accumulator. Pulses in the accumulator were allowed to circulate for about 2 μsec, then removed by applying cutoff bias to the beam-forming electrode of each traveling-wave tube for about 0.1 μsec. Fig. 11 shows some waveform envelopes at various stages of a simple multiplication in the accumulator of Fig. 10.

As it was built, the multiplier had $\tau = 6.25$ nsec and $N = 8$. The shortest interstage delay was about 50 nsec, and the multiplication time 1.6 μsec. Measurements have shown that the minimum delay could be about 12 nsec, and τ could be less than 2 nsec, so it would be feasible to reduce the multiplication time by a factor of four or more.

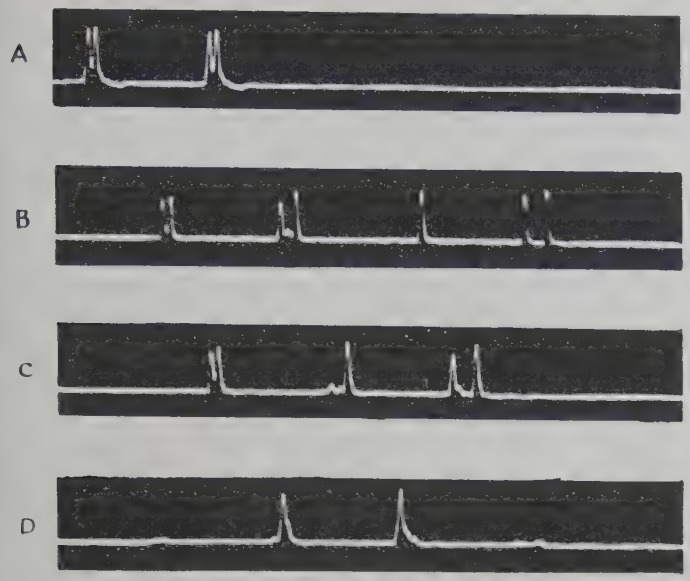


Fig. 11—Waveforms in the accumulator during the multiplication of 3 times 3. Letters refer to test points shown on Fig. 10. The partial products entering the accumulator are shown at *A*. "Carry" signals are shown at *D*, while "intermediate sum" signals appear at *B* and *C*. The last signals at *B* and *C* are the binary digits of the product: 1001.

Conclusion

Pulse rates up to 480 mc have been used successfully with the present logic circuits, and a somewhat higher rate could probably be attained.

The digital multiplier which has been built shows that the basic logic circuit acts as a pulse regenerator and that circulating storage of pulses is feasible. Multiplication at a pulse rate of 160 mc has been demonstrated.

With this computing scheme, signals are delayed by several pulse intervals in traveling-wave amplifiers which must be provided between logic circuits. The logical design of the digital multiplier illustrates that, at least in special cases, such delays may be tolerated while preserving the advantage of a high pulse rate.

Acknowledgment

The work reported here was done in collaboration with G. A. Backman, with the assistance of G. A. Herlich. The logic circuits are based on suggestions of W. M. Goodall and O. E. DeLange. D. W. Hagelbarger devised the arrangement of delays used in the accumulator.

A Logic Design for a Microwave Computer*

STANLEY P. FRANKEL†

Summary—**The properties of presently available components place special emphasis on two desiderata of logic design for use in a microwave digital computer: 1) Smallness of the number of active elements; 2) elimination of information-cycling paths having delay times comparable or short compared with the bit period, as in the conventional flip-flop. A logic design developed in response to these pressures is described in substantially complete detail. Property 1) is obtained by the use throughout of a multiplexing procedure such that the computer functionally (although not physically) resembles a number of nearly identical, and correspondingly slower, computers which are able to operate either independently or in concert.**

I. Introduction

As the electronic computer art has developed during the past few decades, computers of progressively higher speeds have been built. This increased speed has been attained in two ways: by the use of many components and circuits operating in parallel, and by the use of components able to react very quickly. Typical values for the component reaction time have dropped from milliseconds to microseconds and tenths of microseconds. With the development of fast transistors and diodes, modest further reductions of reaction time are in prospect. However, a major reduction in reaction time may be achieved by the use of microwave components—traveling-wave tubes, waveguides, etc. This paper includes a discussion of the special problems which arise from the use of these components in a computer, and the logic design for a microwave computer is developed with attention to these problems.

II. Contextual Discussion

In designing a computer which makes use of microwave components, two pressures are felt to a far greater extent than in conventional computer design. One of these arises from the extremely high cost of all of the components used and of the techniques used in assembling the parts. Part of the high cost is due to the prices paid for traveling-wave tubes, microwave diodes, and other nonlinear elements, and the mazes of plumbing fixtures used for interconnecting; it is also a result of the

* Manuscript received by the PGEC, April 30, 1959; revised manuscript received, July 20, 1959.

† Consultant, General Electric Co., Palo Alto, Calif.; also, 11639 Montana Ave., Los Angeles 49, Calif.

difficulties which arise in assembling these intractable elements and in determining whether or not they are performing as intended. Since these elements are new and relatively little has been done to perfect them, a considerable additional cost arises from the maintenance effort involved in keeping such a system in operation. The pressure of cost makes simplicity of design far more important in this area than in the conventional computers, which are made by stringing diodes and transistors together with simple copper wires.

We may certainly hope that with the passage of time this pressure for extreme simplicity will diminish. As technical skill increases, the cost of each tube wired into a computer should drop and the anticipated interval between replacements should increase. Moreover, after the feasibility and usefulness of a microwave computer have been demonstrated, we may expect increasingly extravagant budgets for succeeding machines. Nevertheless, for a first attempt it seems prudent to seek a design which is far simpler than that of any existing tube or transistor computer.

It is a familiar fact that speed and simplicity are, to some extent, interconvertible in a computer. The simplest computers are one or two orders of magnitude slower in over-all operation than complex machines built of the same components. Since the basic speed attainable with microwave components is several orders of magnitude greater than that which can be attained with conventional parts, it is possible to exchange some of this potential speed increase for design simplicity.

The second pressure involved in microwave computer design arises from an unusual property of the amplifying device which now seems most suitable for use in the microwave domain, the traveling-wave tube. This device amplifies signals over a far broader band of frequencies than can be reached by conventional vacuum tubes. Thus, individual elements of the signal (binary digits of information) may succeed one another at a far greater rate, yet the time consumed in performing the amplification of an element of the signal (the time between its entrance into the amplifier and the emergence of the amplified signal) is not correspondingly shortened in the traveling-wave tube. This delay time of amplification may, in fact, be several times greater than the time interval occupied by each bit. A similar statement may be true for the nonlinear elements necessary for the logical manipulation of information—the gates and buffers by which signals are combined. So far, it is not clear which elements will emerge as the practical devices for the intercombination of microwave signals. These may also prove to involve delays which are long when compared with the attainable bit period. In any case, amplifiers will be required to make good the attenuation of signal strength involved in forming these combinations. The delays introduce a significant new aspect to logic design.

This new aspect may be seen most clearly by considering the microwave analog of the most basic element of a present-day computer, the flip-flop. A conventional flip-flop consists, essentially, of two transistors, each amplifying the signal emerging from the other and inverting its phase. A disturbance introduced from the outside is increased repeatedly as it cycles around the loop provided by the two amplifiers until it has grown to the greatest volume which the amplifiers can accommodate. Thereafter, this saturated signal may be thought of as recirculating indefinitely, with no further change in amplitude. Its effect is to display one of two conditions for the flip-flop as a steady state. In a conventional computer, the bit period, *i.e.*, the minimum time between successive disturbances to the flip-flop, is very much larger than the delay time of a transistor. Thus the disturbance is able to circulate around the loop as many times as is required to reach full amplitude, and to be presented as a temporary steady state, in a modest fraction of the bit period.

A very different situation obtains in a flip-flop formed of traveling-wave tubes which are used as we would like to use them. The loop may contain only one traveling-wave tube and may have, for example, a loop delay of 2 mμsec. Perhaps two circulations would be sufficient to bring the signal up to full strength. Thus, the disturbance may be translated into a steady-state presentation of its information in about 4 mμsec. However, the same tube may have a bandwidth of several kilomegacycles and thus may be able to handle, indivividually, signals appearing at intervals of, say, 1 mμsec. If signals are handled at this full rate of 1 kmc, then the flip-flop would require four bit periods to complete its response to each signal brought to it. Moreover, the bandwidth of the flip-flop circuitry would have to be curtailed to ensure that after 4 mμsec, the signal would be presented continuously, rather than as a series of pulses by 2 mμsec.

III. The Microwave Flip-Flop

It is clear that the flip-flop just described makes inefficient use of the capabilities of the traveling-wave tube. We may compose a more efficient but somewhat different device as follows: Let the traveling-wave tube delay, together with the delay of the recirculation path, be extended to 4 mμsec, and its amplification correspondingly increased, so that one traversal of the amplifier suffices to bring the incoming signal up to saturation. Retain the full bandwidth of several kilomegacycles and use devices to limit signal diffusion so that each signal continues to recirculate within its own 1-mμsec interval. This device will then store and repeatedly present four signals, each one first emerging 4 mμsec after its introduction.

The difficulties involved in using traveling-wave tubes in elements like the classic flip-flop may often be tolerated, since the flip-flop is used as an information storage device and is not always called upon to display the stored content shortly after its reception. Another illustration of the difficulty produced by delays in

amplification can be given in the process of addition. For the sake of simplicity, consider only the addition of binary numbers presented in bit-serial fashion. Each bit of the sum is affected by a carry digit held over from the previous step in the addition. The new carry bit to be stored for use in the succeeding step is a simple logical combination of the old carry bit and the augend and the addend bits. Thus the production of carry bits involves the recirculation of information around a path which includes a simple logic function. For the most straightforward method of addition, this function may be described as follows: the new carry bit agrees with the majority value among the old carry bit, the augend bit, and the addend bit. Simple though it is, the formation and amplitude normalization of this function involve an attenuation of the carry signal which must be made good by an amplifier. Thus, the rate at which the addition can be performed is limited by the delay time incurred in this amplification and function formation, and it may be considerably greater than the bit rate permitted by bandwidth considerations.

Here again, the full bit rate permitted by bandwidth may be exploited if we do not insist that successive bits of the numbers being added appear in successive bit periods. In the intervening bit periods the same equipment may be performing the addition of other pairs of numbers. We are thus led to suggest that each part of a microwave computer operates concurrently on several numbers, instruction words, or other words of information, presented in interdigitated fashion. How many such words must be multiplexed is determined by engineering considerations and need not bear strongly on the logic design. Nevertheless, it is a convenience for the discussion of the logic design to keep some value in mind; therefore we will somewhat arbitrarily assume that this value is four. Numbers will be held in the memory in quadruplexed groups. Each circuit corresponding to the conventional flip-flop holds four bits and presents them in rotation. Therefore, it plays the role of the conventional flip-flop for each of four "quartile computers."

IV. Memory Storage

One of the few things that can be done with microwave signals in a relatively easy way is to delay them. For short delay periods, nondispersive conduction paths, such as coaxial cables, may be used. For long delays, free-space transmission links should be more practical. In either case, a transmission path which delays signals for many bit periods can be used to recirculate a body of information, thus providing a suitable memory element for a microwave computer. In the logic design presented here, only this form of memory is considered, aside from the flip-flop analog. The main memory will consist of a number of long circulating registers, each of which includes a delay path, an amplifier, and gating circuitry. The total delay time for each such circulator is a multiple of a word period in which all of the bits of four words are presented. The number of these circulators and the number of word periods of delay in each circulator are details of engineering which need not be specified in the logic design. Similarly, the number of bits in a word may be left as a free parameter, but to be more definite, it will be taken to be 32 for the purpose of the present description.

In addition to the main memory, two circulators, each of one-word-period delay time, are used. One is the accumulator, which holds four numbers in multiplexed form; the other, the instruction register, which holds four instruction words. The instruction register receives words from any of the main memory circulators; these control the delivery of information to, or reception from, the main memory by the accumulator, and its manipulation by simple arithmetic operations. The operation code of an instruction and the designations of memory circulators selected are held in four-bit flip-flops as described above. Thus, all of the primary avenues for the flow of information handle quadruplexed data, with the four patterns of activity proceeding independently. In first approximation, the computer acts as four completely independent machines, time-sharing the physical equipment but using unrelated information. A few exceptions to this complete separation of the four "quartile" computers are provided. One is the introduction of a multiple-tapped delay line in the circulation path of the accumulator which permits the accumulator of each quartile computer to "eavesdrop" on the other three quartiles. Thus the four quartiles can collaborate in the performance of a single computation. Three bits in the order code, held in flip-flops G, H, and J, control this interquartile communication. (Each of these, like other flip-flops described below, holds a bit for each of the four quartile computers.)

V. Functional Design

Since the cost of providing a circulator for the main memory is nearly independent of its capacity, it is likely that long lines with correspondingly long maximum access times would be used. This can be made compatible with high computing rates by providing for minimum latency programming. For this reason, a one-plus-one address form of instruction word is used, specifying the memory locations of the operand and of the next instruction, in addition to operation code. Six operations are used. They are distinguished by three flip-flops (D, E, and F). The operation codes are shown in Table I.

TABLE I

Order	D	E	F	Effect
Fetch	0	0	0	(M_0) replaces (A)
Add	0	1	0	(M_0) added to (A)
Subtract	1	1	0	(M_0) subtracted from (A)
Extract	1	0	0	(M_0A) replaces (A)
Branch	0	0	1	sign of (A) determines next instruction
Store	1	0	1	(A) replaces (M_0)

Here, the content of the circulator of the main memory selected by the operand address is denoted (M_0). It is formed of the bits, M_0. The accumulator word is (A), of bits A. Similarly, the content of the circulator selected by the next instruction address is denoted M_1.

The operations of addition and subtraction are performed on numbers in binary expansion in the conventional way, with no regard for overflow from the most significant bit position. Numbers so handled may, as the programmer wishes, be interpreted as integers, positive binary fractions, or signed binary fractions. For the last interpretation the most significant bit is assigned the value -1, and the remaining bits are given the values $\frac{1}{2}$, $\frac{1}{4}$, etc. The effect of the branch operation depends upon the most significant bit in the accumulator; that is, on the sign of that number if it is interpreted as a signed fraction.

The order Extract forms the bit-by-bit logical product of the accumulator content with the operand word. The order Store places the accumulator content in memory (as selected by the operand address) while retaining it in the accumulator. These elementary operations suffice for the programming of multiplication and division as well as of more complex arithmetic processes. Multiply and divide are most expeditiously performed by all four quartiles acting in concert.

Each of the two addresses in an instruction word consists of two parts: the designation of one of the memory circulators, which is held in a number of flip-flops (N, O, P, $\cdots$ for the operand; W, X, Y, $\cdots$ for the next instruction); and a "delay number" which indicates the number of word periods which must elapse before the reading of operand or instruction. As an instruction is read from main memory, it is placed in the instruction (circulating) register, R. Thereafter it is recirculated, with successive reduction by one of each of the delay numbers. As each becomes negative, that fact is noted and used to initiate execution or the reading of a new instruction. One or both of the delay numbers may be zero, in which case the execution or instruction reading (or both) occur in the following word period. One exception to this procedure occurs on the order Branch if the accumulator content is found negative; then the next instruction will be read immediately thereafter, with no regard to its delay number.

The initiation of execution is effected by turning on the flip-flop U for one word period. (Again, this statement holds separately for each quartile.) In a similar manner, turning on V causes the reading of a new instruction word into R and the setting of the various flip-flops as indicated. Either U or V or both may be on in successive word periods; in fact, it may be expected that for a well planned problem, both will be on almost continuously, so that an order is executed and another instruction is read in almost every word period.

As each new instruction word is set into R, it must set a number of flip-flops which will influence the following word period (more precisely, a time interval of one word period beginning approximately with the following word period). To accomplish this purpose, a number of taps are provided in the delay line of the R circulator, and bits entering at widely-separated bit periods may be used simultaneously (or nearly so) in setting flip-flops. Bits appearing at these taps are denoted R_1, R_2, etc. Details of timing are also adjusted by the use of short tapped delay lines on the outputs of these flip-flops.

The 32 bit periods of a word period are denoted $a, b, \cdots, y, z, \alpha, \beta, \gamma, \delta, \epsilon, \zeta$, and are marked by signals produced by a tapped circulator provided with a Cutler circuit which ensures circulation of a single pulse. Specifically, each of the signals (a, b, etc.) is present for four successive bit periods, one for each quartile computer. These signals (and their inverses) are used in the logic design to single out the bit period in which a flip-flop is to be set or some other activity accomplished. They are also used in the input and output mechanisms which serve to translate information from microwave signals to the much slower representation used in external circuitry and vice versa. These input, output, and auxiliary devices will not be further described here.

Two flip-flops play a more active role in the computer than those described above. One, denoted C, holds carry (or borrow) bits in the execution of the arithmetic operations addition and subtraction. The other, K, holds borrow bits occurring in the reduction of the delay numbers circulating in R. Each of these may change from one (quartile) bit period to the next, with each new state dependent on the prior state. It is in these two circuits that the amplifier delay time applies the most stringent conditions. In other parts of the logic circuitry of the computer, signals enter from, and go off to, delay lines of considerable length, so that amplifier delays may be accommodated by suitable adjustment of these long delays.

VI. Logic Design

The logic design is shown in summary in Table II. Here conventional Boolean algebraic notation is used (overscore for complementation, $+$ sign for *inclusive* OR, and inequality sign for *exclusive* OR). The notation is convenient by reason of its familiarity. Its choice is not, however, meant to imply that these logic functions are the easiest to mechanize or lead to the most compact expressions for the design. For example, the microwave mechanization of the logic for the carry flip-flop input would probably use a simple majority-of-three element, here represented by a complex parenthetical expression. Since each flip-flop, as well as A or R, is in fact a circulating register, its input is specified in single-input form and is indicated by a prime following the name of the circulator.

A. Delay Number Count-Down

An instruction word is set into R during a word period marked V, then recirculated during word periods marked $\overline{V}$. More precisely, the signal presented by a suitably delayed version of the signal V (a tap on the delay line emerging from V) is used. This delayed signal is denoted V_r. The information to be inserted into R is thus described by

$$R^* \equiv (V_r M_0 + \overline{V}_r R), \tag{1}$$

where M_0 is the operand memory signal as selected by flip-flops N, O, P, $\cdots$. At the end of the word period into which this instruction is introduced, the part of its information which will be needed in static form is set into the flip-flops G, H, J, D, E, F, etc., as shown by the last equations in Table II. The various taps in the R circulator (R_{31}, R_{30}, etc.) are used to permit setting these flip-flops approximately simultaneously.

The part of the instruction word which enters R during the bit periods g to m is the delay number of the operand. It is reduced by one at the time of its introduction and on each subsequent circulation. This reduction is accomplished with the help of flip-flop K, which holds borrow bits for this reduction (separately for each quartile). It is turned on at time f, *i.e.*, just prior to the appearance of the operand delay number, and remains on as long as zeros appear in the number to be reduced, R^*. For this purpose, the input to the K flip-flop might be written as $f+K\overline{R}^*$. The reduction of the number formed of the bits R^* is then accomplished by complementing R^* as long as the borrowed condition, indicated when K is on, persists. Since a similar reduction of the next-instruction delay number, appearing in the bit periods t to z, is to be performed, the borrow flip-flop is also turned on at time s in preparation for this later reduction. The full equation for the K flip-flop becomes,

$$K' = f + s + K\overline{R}^*. \tag{2}$$

The reduction of the delay numbers in the instruction introduced into or circulating in R is described by

$$R' = (K \neq R^*). \tag{3}$$

The reduction of the operand delay number past zero is detected by the ON state of K at time n; this circumstance is used to turn on flip-flop U, which will initiate execution of the order. This is shown by:

$$U' = \bar{n}U + nK. \tag{4}$$

Similarly, if K is on after the transmission of the next instruction delay number, *i.e.*, at time α, V is turned on to produce the reading of a new instruction. This is shown by:

$$V' = \bar{\alpha}V + \alpha K + \cdots. \tag{5}$$

B. Branch Order

The branch order, indicated by the operation code $\overline{DE}F$, causes immediate reading of a new instruction if the most significant bit held in the accumulator is a one. This bit may be read at time α of the execution word period from a suitable tap in the accumulator delay line, designated A_b. This occasion for reading a new instruction is indicated by an additional term placed in (5). The full equation is now:

$$V' = \bar{\alpha}V + \alpha K + \alpha\overline{DE}FUA_b. \tag{6}$$

The precise location of this tap is selected by the flip-flops G, H, and J, which read bits of the order code. This permits the branch instruction in each quartile computer to use the accumulator content from other quartiles as well as its own. The selection of the tap delivering A_b is described by:

$$A_b = \overline{GHJ}A_0 + G\overline{HJ}A_1 + \cdots. \tag{7}$$

This same tap selection process may be used for the purpose of interquartile traffic in the Store order, in which the accumulator output (suitably delayed from A_b) is placed in a selected memory location. Still another use for this tap selection is made possible by the inclusion of the accumulator itself among the circulators from which information may be received, either as M_0 or as M_I. In this way, the accumulator contents of two quartiles may be combined by addition, subtraction, or extraction.

C. Arithmetic Orders

In addition and subtraction operations, the recirculated accumulator content, A, is combined with a number brought from the selected operand memory source to produce the accumulator input. In this process, use is also made of the carry-borrow flip-flop C, which is set to zero prior to each word period and set to one in suitable circumstances thereafter. The rule for addition is a widely familiar one: the new carry bit, C, is the majority among A, C, and M_0, while the sum bit is formed by the exclusive OR combination of these three. This rule is expressed algebraically as:

$$\begin{aligned} C' &= \bar{\zeta}(AC + AM_0 + CM_0) \\ A' &= (A \neq C \neq M_0). \end{aligned} \tag{8}$$

The qualifying factors to limit these terms to the addition process have been omitted here. The rule for subtraction is obtained from this by complementing A as it occurs in the C' equation. Since the flip-flop C is not used for any other purpose, its full equation becomes:

$$C' = \bar{\zeta}[CM_0 + (C + M_0)(A \neq D)]. \tag{9}$$

The state of flip-flop D serves to distinguish addition and subtraction orders.

The above expression for A' will hold whenever an addition or subtraction is being executed, *i.e.*, in the circumstances $E\overline{F}U$. Thus one term in the accumulator input equation is:

$$A' = E\overline{F}U(A \neq C \neq M_0) + \cdots . \quad (10)$$

The remaining terms in A' are simple. In executing the Fetch order, M_0 is set into the accumulator; while on Extract, the product AM_0 is inserted. In all other circumstances, A is recirculated unchanged. Thus altogether,

$$A' = A(F + \overline{U}) + \overline{F}U[E(A \neq C \neq M_0) + \overline{E}M_0(A + \overline{D})]. \quad (11)$$

D. Control Flip-Flops

The flip-flops controlling memory circulator selection (N, O, $\cdots$, and W, X, $\cdots$) and those holding the operation code (G, H, J, and D, E, F) are set more or less simultaneously by the instruction word in the register R. (It would be logically sufficient, but less convenient, to confine this setting process to the word period marked V in which the instruction is brought into R.) For definiteness of description, the time of this setting will here be called ω, and should be understood to occur near the end of a word period. Then the setting of flip-flops is described by:

$$G' = \omega R_{31} + \bar{\omega}G$$
$$H' = \omega R_{30} + \bar{\omega}H$$

etc., to

$$Z' = \omega R_1 + \bar{\omega}Z. \quad (12)$$

The number of these flip-flops, the placement of the corresponding bits in the instruction word, and the corresponding placement of taps on the R delay line are free parameters in the logic design, to be fixed on the basis of engineering considerations.

E. Memory Circulators

Each of the memory circulators recirculates its content without change except when it is selected by the operand flip-flops N, O, P, $\cdots$, during the execution of a Store instruction. Then the input to the circulator is a suitably delayed version of A_b, which will be denoted A_d. (The word so delivered to memory is one of the four quartile accumulator contents, as selected by by G, H, and J.) A typical equation for a memory circulator input is thus

$$M_3' = (UD\overline{E}FN\overline{O}P \cdots)A_d + (\overline{U} + \overline{D} + E + \overline{F} + \overline{N} + O + \cdots)M_3. \quad (13)$$

The input and output devices may be addressed as if they were memory circulators. To facilitate the initial filling of the microwave computer memory (the so-called "bootstrap" process), they are given the easily distinguished designation, $\overline{NO}\overline{P}\cdots$ and may thus be thought of as being a circulator, $M_{(0)}$. The bootstrap process makes use of a device, not previously described, to set to zero all bits entering the instruction circulator. On release of this inhibition, the next instruction will be read from the input device, $M_{(0)}$.

Conclusion

The logic design presented above is summarized in Table II. It should be borne in mind that these equations are inexplicit in many ways, particularly with respect to the handling of the small time delays suffered by a signal in its progress through the logic.

TABLE II

$K' = f + s + K\overline{R}^*$
$R' = (K \neq R^*)$ where $R^* = V_r M_0 + \overline{V}_r R$
$U' = \bar{n}U + nK$
$V' = \bar{\alpha}V + \alpha K + \alpha\overline{DE}FUA_b$
$A_b' = \overline{GHJ}A_0 + G\overline{HJ}A_1 + \cdots$
$C' = \bar{\zeta}[CM_0 + (C + M_0)(A \neq D)]$
$A' = A(F + \overline{U}) + \overline{F}U[E(A \neq C \neq M_0) + \overline{E}M_0(A + \overline{D})]$
$G' = \omega R_{31} + \bar{\omega}G$; etc.
$M_1' = (UD\overline{E}FN\overline{O}\overline{P} \cdots)A_d + (\overline{U} + \overline{D} + E + \overline{F} + \overline{N} + O + \cdots)M_1$; etc.

Acknowledgment

This work was made possible by stimulating discussions with L. P. Elbinger and by his encouragement and criticism.

Parametric Phase-Locked Oscillator—Characteristics and Applications to Digital Systems*

L. S. ONYSHKEVYCH†, W. F. KOSONOCKY†, AND A. W. LO†

Summary—The ability of the Parametric Phase-Locked Oscillator (PLO) to detect, amplify, and store binary digital signals, in the form of two distinct phases of a carrier, makes it possible to use the device as the sole component in a digital computer system. The variable-capacitance version of the device operates readily at kilomegacycle frequencies, thus forming the basis of a digital computer at a kilomegapulse clock rate.

In the present paper the results of an investigation of the behavior and possible applications of the variable-capacitance PLO are presented. The investigation was supported by experimental work with lumped-component variable-capacitance PLO's at 5 mc, and microwave variable-capacitance PLO's at 4 kmc.

The steady-state behavior of the device is described; variations of the output voltage with pump voltage, loading, tuning and frequency variations are presented in the form of characteristic curves. Results indicate that the device is rather insensitive to reasonable changes in operating conditions and parameter values.

The transient behavior of the PLO shows that the device can be switched in a number of different ways. Five such modes of operation are discussed; these are phase initiation, forced switching, burst generation, tri-stable operation and unconditional switching. Each of these modes has particular advantages for various applications. Switching times of the order of 3 to 10 cycles of the signal frequency are readily obtainable.

The various modes of operation of the device suggest a number of applications both in logic and in memory. To illustrate the versatility of the device, a random access memory is described as an example.

Introduction

THE principles governing the operation of the Parametric Phase-Locked Oscillator (PLO) have been described by von Neumann [2] and independently developed by Goto of Japan [1]. The ability of such a device to perform all the basic functions—discrimination, amplification, and storage—needed in a digital computer has been well illustrated by digital systems built with parametrons in Japan [1].

The parametron, a variable inductance version of the PLO, is limited to medium operating frequencies because of hysteresis loss in the ferrite cores. Some variable-capacitance devices free from hysteresis loss have been suggested for such an application [2], [5]. Indeed, the variable-capacitance PLO has been operated at very high frequencies, permitting digital computation at kilomegapulse rates [3], [4].

This paper reports the results of an investigation by the authors of the steady-state and switching behavior of the variable-capacitance PLO. Various modes of operation are described; their applications to logic and memory systems are illustrated. To facilitate instrumentation, most of the detailed experimental work was performed with a lumped-parameter PLO, at an operating frequency of 3–10 mc. The pertinent properties of the device were experimentally verified with a distributed parameter PLO operated at 2—4 kmc. The results indicated that the lumped-parameter PLO is a good scaled-down frequency model of the microwave PLO.

* Manuscript received by the PGEC, April 30, 1959. Presented at the ONR Symp. on Microwave Techniques for Computing Systems, Washington, D. C., March 12, 1959.

† RCA Laboratories, Princeton, N. J.

Principle of Operation

The Parametric Phase-Locked Oscillator consists essentially of a tank circuit tuned to a natural resonant frequency f_N. If one of the reactances is made to vary at a frequency f' by an energizing source, often referred to as the "pump," then an effective negative resistance at frequency f, where $f=(n/2)\,f'$ $(n=1, 2, 3, 4 \cdots)$, appears in the circuit. If $f \cong f_N$ the circuit will start to oscillate "parametrically" at this frequency [see Fig. 1(a)]. The parametric oscillations are "locked" in phase

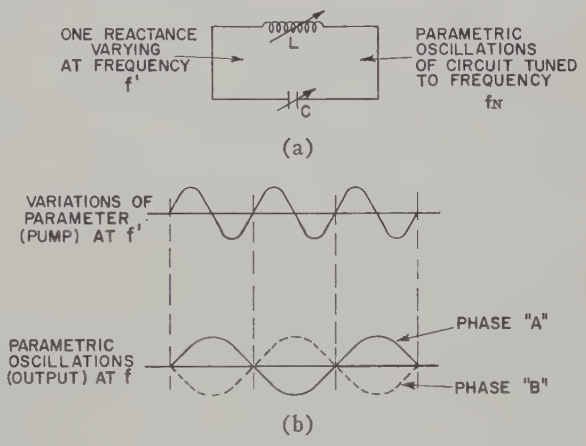


Fig. 1—Parametric phase-locked oscillator: (a) basic PLO circuit, (b) relations between output phases (for case of $f=2f$).

to the oscillations of the pump. Furthermore the sustained oscillations can occur in either one of two possible phases 180° apart [phases A and B in Fig. 1(b)] which are used to carry binary information. Once the circuit starts to oscillate in either one of those two phases, it continues to do so, until forcibly stopped or changed. If the circuit is initially at rest and then the pump is suddenly applied, oscillation builds up; either of the two possible phases is equally likely to occur. Which one actually does occur is determined by initial conditions; that is, by random noise in the absence of any signal. We can, however, steer the circuit into one or the other phase by applying to the circuit a small signal, of frequency f and the desired phase, during the time when

the oscillation is starting to build up. This input signal is referred to as the "locking" signal.

In the "conventional" mode of operation, the value of n in the equation above is 1, so that the pump is at twice the frequency of the parametric oscillation. This paper deals primarily with this mode of operation (because this mode is the most efficient one, as will be explained more fully later in the report), except in certain places where it is clearly noted otherwise.

The mathematical theory of operation of the PLO has been reported in other papers [1], [2], [5]–[7], so that none of the mathematical treatment of the behavior of this device is presented here. This paper contains mainly experimental results obtained by work with actual devices.

The circuit used in the 4–10 mc experiments is shown in Fig. 2(a). The nonlinear capacitance consists of the two back-biased junction diodes. Two diodes are used in order to balance the circuit, so that the pump frequency does not appear at the output terminals and vice versa. 1N93 diodes were used in the low-frequency experiments.

For microwave experiments at 2–4 kmc, a circuit described by Sterzer and Beam [3], [8] was used. The microwave PLO circuit is shown in Fig. 2(b). It consists of a variable-capacitance diode in a resonant cavity. Microstrip is used for the cavity, as well as for input-output and pump connections. Filters were used to separate the signal and pump frequencies. Special RCA variable-capacitance point-contact diodes were used in the experiments. For a more complete discussion of the microwave circuit the reader is referred to the papers by Sterzer and Beam [3], [8]. A symbolic representation of the PLO is given in Fig. 2(c).

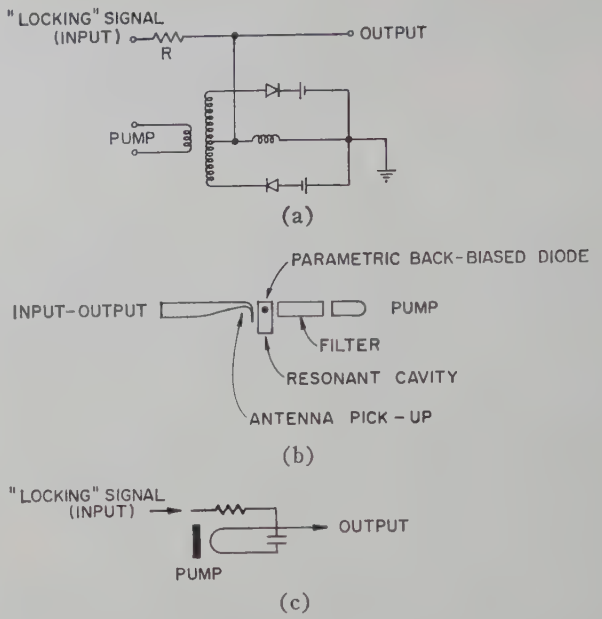


Fig. 2—Parametric phase-locked oscillator circuits: (a) one possible lumped-parameter variable-capacitance PLO circuit, (b) one possible microwave PLO (in microstrip), (c) symbolic representation of a PLO.

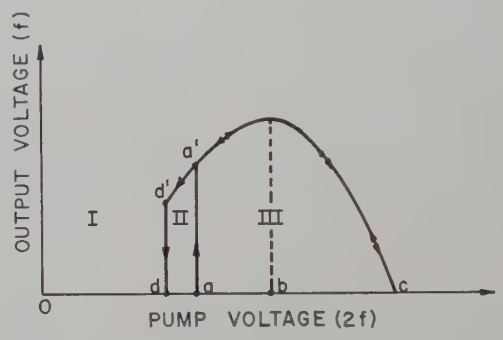


Fig. 3—The characteristic curve (output voltage vs pump voltage).

Steady-State Behavior

The output voltage of a Parametric Phase-Locked Oscillator is plotted as a function of the pump voltage, to obtain a steady-state characteristic curve as shown in Fig. 3. This curve shows that the PLO has three regions of operation. In Region I, the PLO does not oscillate at all, so that the output voltage is zero. In Region II, the PLO can either oscillate or not, depending upon previous history. (There is a pseudo-hysteresis loop. The oscillation can be in either of two phases, so that three stable conditions are possible: oscillation in phase A, oscillation in phase B, or no oscillation at all. This region we may call the tri-stable region.) In Region III, the PLO invariably oscillates in either phase A or phase B. This region is called the bistable region.

When the pump voltage is increased from zero, at first no output is observed until point a is reached. At this point the device starts to oscillate in either of the two possible phases. The particular phase into which the device locks itself is determined by noise, or by a purposely introduced "locking signal."

When the pump voltage is increased further (beyond point a) at first there is an increase in the output voltage until point b is reached, where the output is at its maximum. Increases in pump voltage beyond this point cause excessive diode conduction, thus loading the circuit, with resulting decrease in output amplitude, until finally point c is reached where the output goes to zero and oscillation stops. If the pump voltage is now reduced, the previous path is retraced until point a is reached. At this point the oscillation does not die out but continues to point d where oscillation stops.

The microwave PLO behaves in a similar manner. This is illustrated by the steady-state characteristic curve of a microstrip PLO operated at a frequency of 4 kmc, shown in Fig. 4. Sometimes there may be no tri-stable region.

The characteristic curve of Fig. 3 is typical and was plotted for a particular value of pump frequency and tuning. The curve would be slightly different if either the pump frequency or the tuning of the unit were changed.

Let us first look at the effects of changing the pump frequency leaving all parameters of the PLO unchanged. If the frequency is lowered the characteristic curve shifts

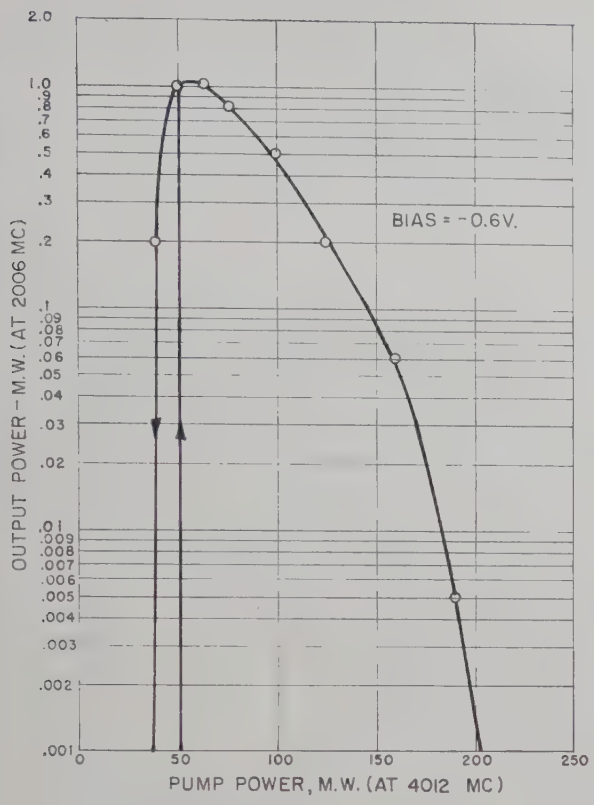


Fig. 4—Characteristic curve of a microwave PLO.

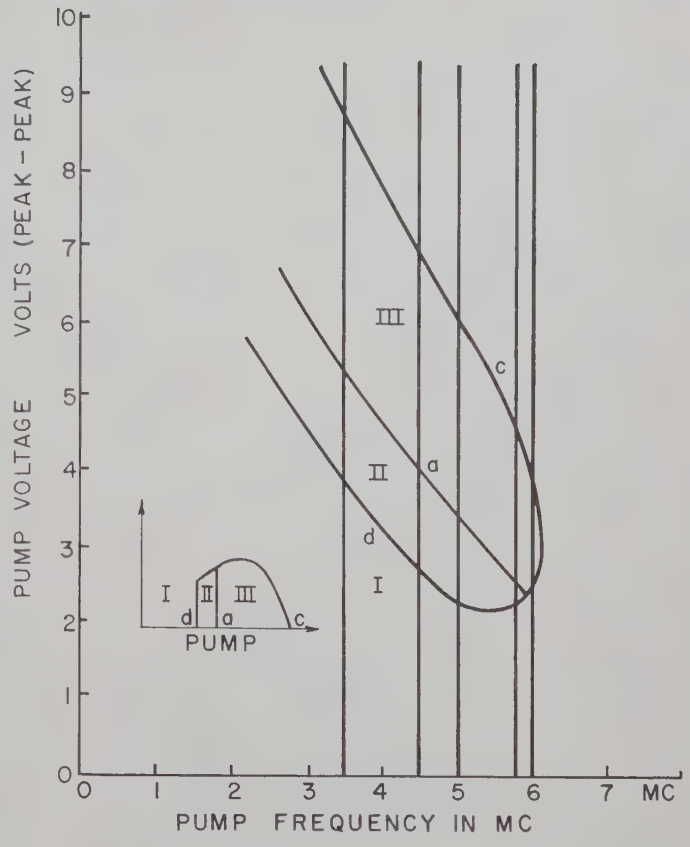


Fig. 5—Frequency characteristics.

to the right, and it takes more pump voltage to operate the device. If the frequency is increased, all the points shift to the left; also points a and d come closer to each other, so that the pseudo-hysteresis loop becomes thinner, until finally it disappears completely and there is no snap action at all. The output amplitude decreases, until finally the frequency is too high to sustain oscillation at any value of pump voltage.

In order to see these variations more clearly, we plot pump voltage at points a, d, and c against pump frequency and obtain a "frequency characteristic" as seen in Fig. 5. This plot shows the bi-stable and tri-stable regions of operation in the pump amplitude-pump frequency domain.

Changing the natural resonant frequency (f_N) of the PLO amounts to shifting the frequency characteristic curve along the frequency axis. Experiments indicate that changing f_N is equivalent to changing the pump frequency; that is, an increase in f_N is essentially interchangeable with a decrease in pump frequency and vice versa. For this reason, experimental data were obtained by varying either f_N or the pump frequency, whichever was more convenient.

In the vicinity of the natural frequency (f_N) the oscillation of a PLO can be sustained with minimum pump amplitude (see Figs. 5 and 19). The output of the unit depends on the difference between the driving (pump) frequency and the natural frequency (f_N) and not on the pump frequency or f_N alone.

All of the preceding discussion applies to the conventional mode of operation, the "2 to 1" mode, with the pump at twice the frequency of the output signal. It is possible, however, to obtain other modes: "1 to 1" (with pump and output signal at the same frequency), and even "1 to 2" and "2 to 3" (with the output signal at a higher frequency than the pump).

The regions of operation for these various modes were mapped and are shown in Fig. 6. For the particular nonlinear capacitance used in our circuits, the conventional 2 to 1 mode has the widest region of operation, is easier to work with, and is also more efficient than the other modes.

The effect of loading upon the steady-state behavior of a PLO is shown in Fig. 7. The most important effect is the decrease in peak output amplitude. We can plot the peak output amplitude vs the loading, as in Fig. 8; this plot indicates the maximum loading capacity of any particular PLO unit. (Maximum loading was taken as that loading for which the output amplitude decreased to 80 per cent of its unloaded value.) Loading curves for the four different modes are shown. These characteristics were taken for optimum values of pump frequency and amplitude. As expected, the conventional 2 to 1 mode provides the best loading possibilities.

The investigation also includes an experimental study of the effects of the amount of nonlinearity of the variable reactance (capacitance) on the steady-state behavior of the device. The results are summarized in Figs. 9 and 10, where we see the variation of width of region of operation (over which 2 to 1 oscillations occur)

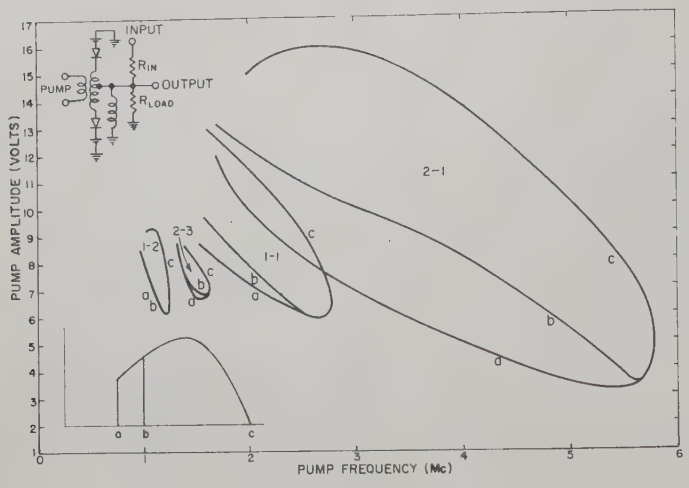


Fig. 6—Regions of operation.

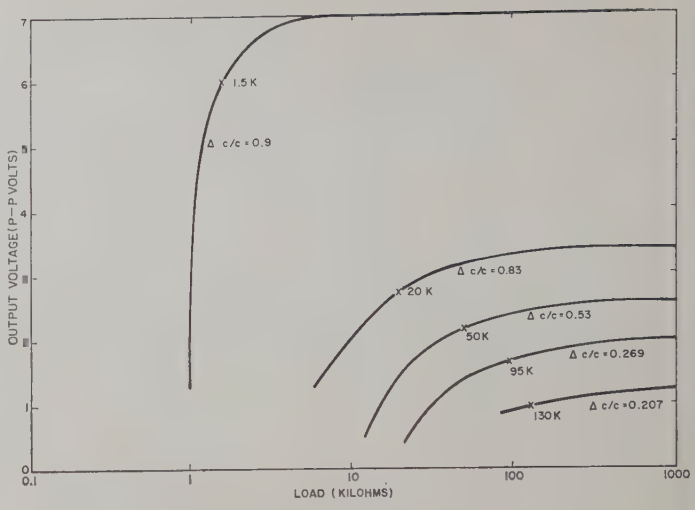


Fig. 9—Output voltage vs load with $\Delta C/C$ as the running parameter.

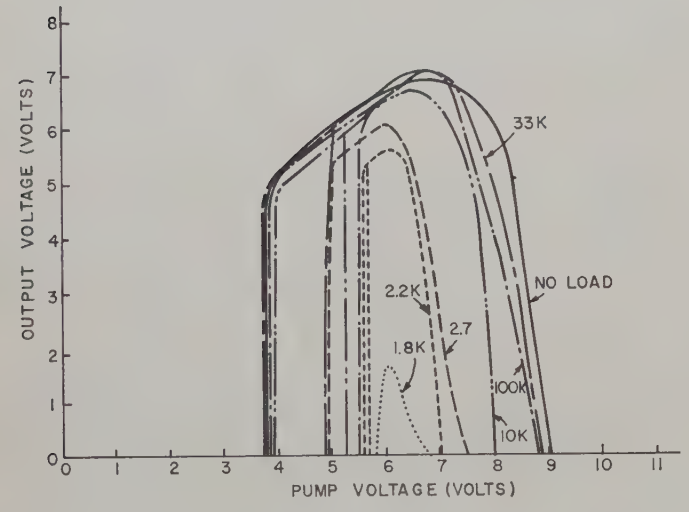


Fig. 7—Characteristic curves for various values of loading (at 3.5 mc).

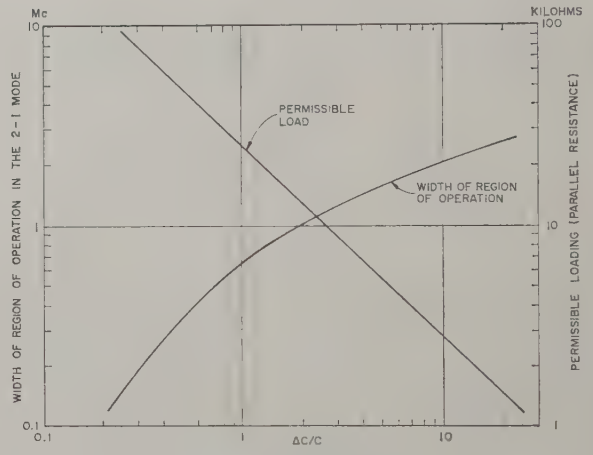


Fig. 10—Effect of $\Delta C/C$.

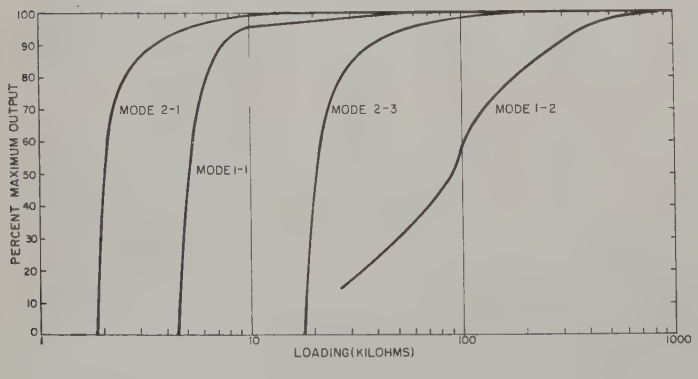


Fig. 8—Maximum loading characteristics.

and maximum loading with the amount of variation of the capacitance ($\Delta C/C$) change of capacitance divided by the capacitance at the quiescent point. It is assumed that at maximum output amplitude the pump voltage swing just starts to enter the conduction region.

From the graph of Fig. 10 we see that with an increase in change of capacitance $\Delta C/C$, we get an increase in both width of region of operation and maximum loading. Therefore it is very desirable to obtain diodes with as much nonlinearity as possible. A small degree of nonlinearity gives a smaller permissible operating frequency range, lower efficiency and a lower loading capability; on the other hand, the output voltage waveform deviates less from a sinusoidal wave.

The above data on the steady-state behavior of the PLO were taken at pump frequencies of about 5 mc. Similar results were obtained with a microwave PLO at 4 kmc. In general, very good correlation was observed between the behavior of a 5-mc PLO and the behavior of a 4-kmc PLO.

Switching Behavior

It has been shown in the previous sections that a PLO can oscillate in either one of two possible phases. It is locked to one or the other of these two phases and sustained oscillation with intermediate phase is impossible. These two phases can be used to carry and store binary information. This section of the paper deals with the ways of changing the phase of an oscillating PLO and the ways of starting the PLO in the desired phase.

There are several different methods of determining or

changing the phase of a PLO: 1) forced switching by injecting a signal of the proper phase; 2) locking the starting oscillation to an injected ("locking") signal; 3) momentarily interrupting the parametric oscillation for a fixed length of time; and 4) triggering oscillation in the tri-stable mode. The various methods are more fully described below.

Forced Switching

Forced switching is probably the most straightforward method of changing the phase of a PLO. It consists of injecting a large amplitude signal of frequency f and of the desired phase into an oscillating PLO unit and thus forcefully determining the phase of oscillation. The injected input signal is called the "locking signal." If the PLO is in the opposite phase from that of the locking signal, the oscillation is first quenched and then started in the new phase. If the two phases agree, no change in the output occurs, except that the amplitude of the output signal may increase slightly for the duration of the input signal if the output is not already at its maximum.

The above discussion refers to the bi-stable region (III) of operation. In the tri-stable region (II) the locking signal has the following effect: if the PLO was not originally oscillating, the locking signal may trigger it into oscillation; if it was oscillating in the phase opposite to that of the locking signal, the oscillation is first quenched and then, if the locking signal is strong enough and long enough, oscillation will start in the phase of the locking signal.

The switching speed of the PLO (at a particular value of loading and tuning) is illustrated by the curves of Fig. 11. These are curves of constant switching time for given coupling as a function of locking and pump voltages. Switching time is defined as the time from the start of the locking signal to the point where the oscillations are in the new phase at full amplitude.

For memory applications it would be desirable to operate the PLO in "coincidence"; that is, to switch the phase of oscillation if, and only if, all of two or more inputs to the device are selected. To obtain this, one of the inputs (or even all but one) acting singly should not be large and/or long enough to cause a phase change but all of the inputs acting together should do so. There are two ways to obtain such coincident operation by forced switching; either by coincidence of two locking signals, or by coincidence of a locking signal and pump modulation.

From the curves of forced switching it is easily seen that a two-to-one selection by voltage addition of two locking signals is rather easy and fast. For example (see Fig. 11), at a pump voltage of 5.5 volts, a 6-volt locking signal cannot produce switching even in 25 cycles, but if two such 6-volt signals are applied in coincidence they provide a total locking signal voltage of 12 volts, thus switching the oscillation in less than 5 cycles.

Another way to accomplish coincident switching consists of applying a locking signal, while at the same time

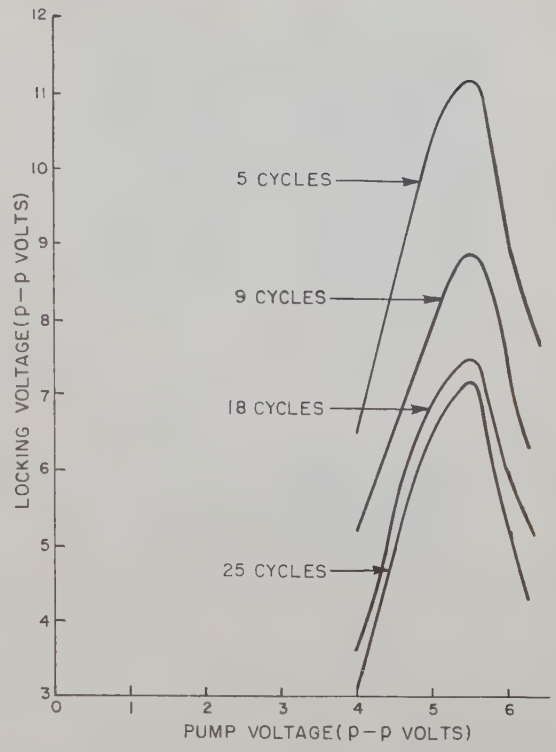


Fig. 11—Forced-switching characteristics (at 4.5 mc, $R_{in}=10$ K).

changing the amplitude of the pump. To explain, let us assume that the normal operation of the PLO is at 5.5 volts pump voltage. This means that introduction of a locking signal at 7 volts for even as long as 25 cycles will not produce switching. However, if, while the locking signal is introduced, the pump voltage is reduced to 4.5 volts, the PLO changes phase in approximately 7 cycles, provided that the locking signal was of phase opposite to that of the oscillator.

Fig. 12 is a photograph of the locking signal-pump modulation coincident scheme; the phase of oscillation of a PLO is switched in approximately 4.5 cycles at a certain value of pump voltage, but if the pump voltage is increased slightly, the same locking signal produces no switching.

Comparing the two modes of coincident operation, it is found that the coincidence of two locking signals is slightly faster and less critical, but both of the methods can be made to work reliably and perform satisfactorily.

Phase Initiation

Switching by phase initiation can best be described by reference to Fig. 13. At time t_0 the pump voltage has been removed and the PLO is not oscillating. At time t_1 we apply a locking signal through some form of loose coupling (high resistance, transformer, or antenna), and at t_2 the pump starts. The PLO starts to oscillate at time t_2 in the same phase as the locking signal; the phase of the oscillation is locked to the small locking signal during the build-up of oscillation. (The voltages are not shown to scale.) The energy of the locking signal actually

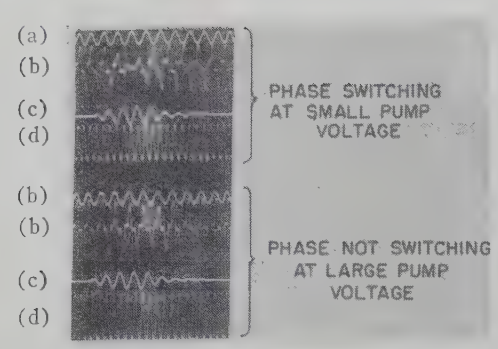


Fig. 12—Coincident write-in: (a) reference, (b) output signal, (c) input signal, (d) pump.

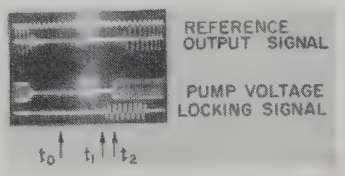


Fig. 13—Phase initiation.

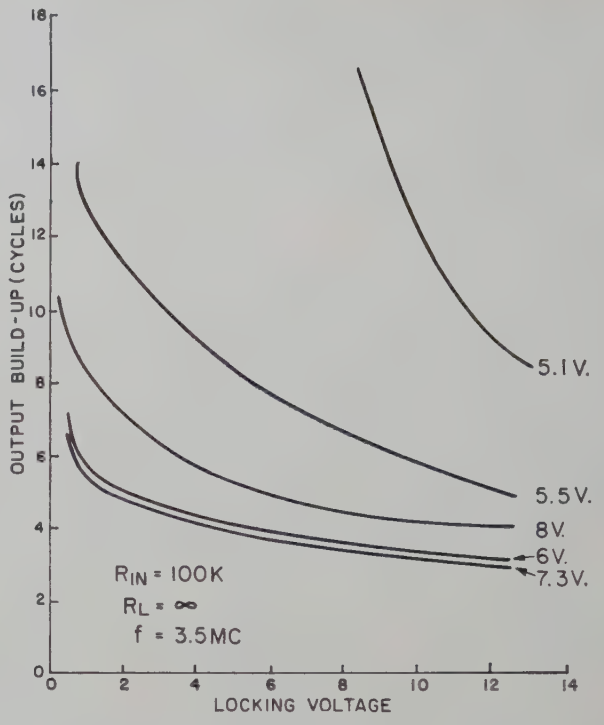


Fig. 14—Phase locking characteristics (at 3.5 mc and no loading, with pump voltage as the running parameter).

coupled to the tank is much lower than the energy of the tank oscillation.

The questions of interest in this mode of operation are the time it takes for the device to build up from zero for various driving and loading conditions, the amount of input needed, and the amount of loading that can be tolerated.

The family of curves of Fig. 14 shows the variation of build-up time with locking signal voltage, with pump voltage as the running parameter. The total build-up time plotted in these curves consists of two parts: an apparent delay, when the device appears not to oscillate at all, even though both locking and pump voltages are present; and a pronounced rise time (to the final value). Increasing the locking signal amplitude will reduce both the delay time and the rise time; however, the effect on the delay time is more pronounced. For large amplitude locking signals the delay disappears.

On the basis of a large number of experimental curves of the type shown in Fig. 14 the following conclusions were reached.

1) The build-up time is shortest if the pump amplitude is such that maximum output results. Thus the build-up time is directly related to the output voltage.

2) Loading (lowering of the Q) slows down the rise time. On the other hand, loading speeds up the time for oscillation to decay when the pump voltage is removed. (Loading consists of the composite effect of input and output couplings.)

Phase initiation as a means to perform logic has been developed by Goto and described in [1]. This method is entirely applicable to the variable capacitance PLO at microwave frequencies. A brief discussion of this method is included here.

Information is easily transferred between two PLO units by using the phase initiation process. The two units are coupled loosely. The first unit is oscillating; its phase holds binary information. The second unit is originally not energized. When the pump for the second unit is turned on, the unit starts to build up an oscillation in a phase dictated by the phase of oscillation in the first unit. The PLO, however, is basically a two-terminal, bilateral device. Therefore, in order to obtain transfer of information in a specified direction only, an overlapping sequence of three pump sources is used. The operation is best described with reference to the PLO Shift Register in Fig. 15.

Initially only PLO No. 1 is oscillating in phase A. Then, PLO No. 2 is energized by a pump burst; through the loose coupling by the phase initiation process, it starts to oscillate in phase A. PLO No. 1 now ceases to oscillate. Information has been transferred in the forward direction from unit 1 to unit 2 but not to units 0 and 3 since they were not energized. In the next transfer, units 1 and 4 will be inactive, so that information can be transferred from unit 2 to unit 3.

Using this type of transfer mechanism, logical gating can be achieved by linear addition of input signals (majority logic). To obtain the conventional AND and OR gates a bias input at a reference phase is used. At low frequencies inversion is achieved by transformer coupling; at microwave frequencies it is achieved by appropriate phase shifting.

This type of logic has been extensively developed and proven in actual large operating systems in Japan [2]. There are no particular difficulties in adapting phase initiation logic to microwave PLO systems. Phase initiation has been studied at 4 kmc pump frequency. It was found that rise times of 8 cycles are easily achievable.

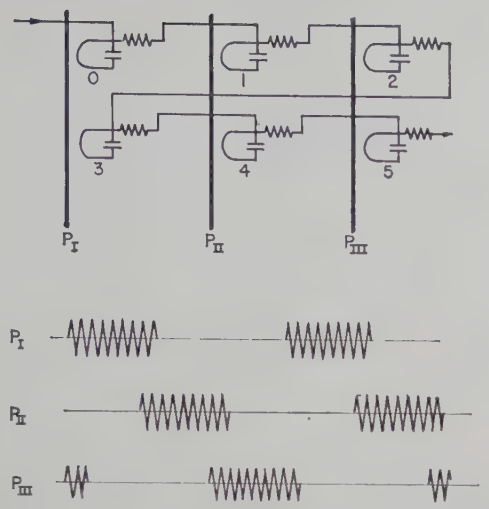


Fig. 15—Shift register with the three-phase pump system.

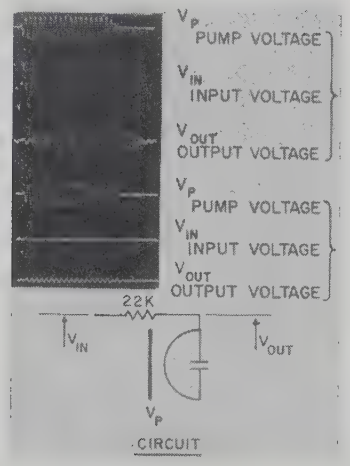


Fig. 16—Burst generation. Scale: pump—5 v/cm, input-output—2 volts/cm, and time 1 μsec/cm.

In some experimental work at microwave frequencies the turning off of the PLO was accomplished by pulse biasing the diode out of the oscillating region, rather than by shutting off the pump. Both methods are feasible. It is foreseeable that at microwave frequencies the 3-clock system described above may be abandoned if the use of isolators and directional couplers to obtain transfer in the desired direction can be made practical.

Burst Generation

If the pump is suddenly applied to a PLO originally at rest, there is an appreciable delay time before the unit starts to oscillate. This delay time disappears if a sufficiently large locking signal is present (see the section of this paper on phase initiation). This behavior of the device forms the basis of a new mode of operation, in which the pump is normally off, except for short periods of time ("bursts"). The operation is adjusted in such a way that the delay in starting oscillation in the absence of a locking signal input is longer than the duration of the pump burst. Therefore, in the absence of a locking signal, no output will result. If, on the other hand, there is a locking signal during the pump burst, an output will result, since the delay will disappear and oscillation appears soon after the pump is applied. This mode of operation is shown in the photographs of Fig. 16.

Various gates and logical circuits can be constructed using the burst generation method. For example, in Fig. 17 a selection switch is shown. The units are biased by a reference phase (phase 1). Only the units which receive a 1 input will produce a burst output, as shown. The other units receive effectively no input (since a 0 input is cancelled by the bias); hence, no burst output is produced.

Unconditional Switching

The Parametric Phase-Locked Oscillator can be unconditionally triggered from its original phase to the other phase by simply interrupting the forced oscillation for a definite length of time.

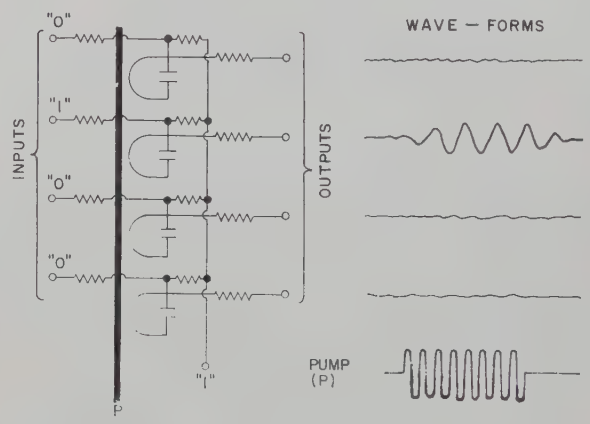


Fig. 17—PLO selector switch using "burst generation."

This action can be explained in the following manner [see Fig. 18(a)]. When the parametric oscillator is taken out of phase-locked oscillation (either by a change of the pump amplitude out of the operating range, or some other cause), there is still energy stored in the reactive components of the tank circuit. Therefore, the tank continues to oscillate, for a length of time, *at its natural frequency* (f_N). The natural frequency is usually slightly different from the frequency (f) at which the pump drives the unit.

If the conditions for oscillation are restored before all of the energy in the tank circuit is dissipated, the PLO will again start to oscillate parametrically. The phase of the oscillation will be determined by the remaining small oscillation still present in the tank circuit when the conditions for oscillation are restored.

If the length of the interruption is such that *an odd number of half cycles* is lost or gained, the oscillations change phase; *i.e.*, the unit starts to oscillate in the phase opposite to the phase in which it was originally oscillating. If the number of half cycles is even, no such change in phase takes place. It was found that for most reliable operation the interruption should be such that only one half-cycle is lost or gained. Otherwise the oscil-

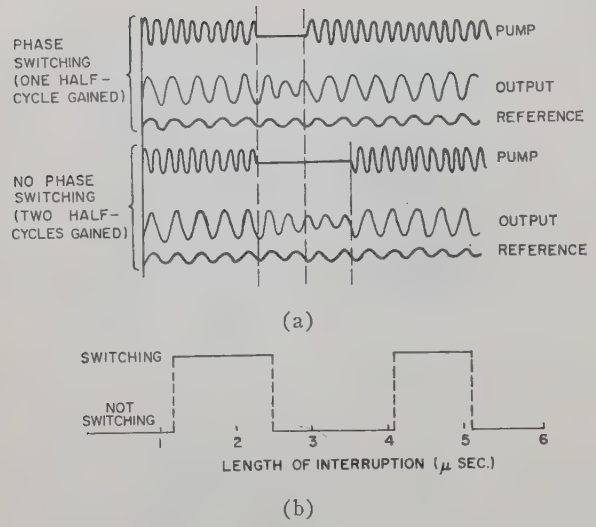


Fig. 18—Unconditional switching: (a) waveforms in unconditional switching, (b) regions of switching.

lations tend to become so small that they start to approach the noise level, and operation becomes unreliable.

In Fig. 18(b) a typical switching diagram for unconditional switching is shown. In this experiment the interruption was obtained by completely shutting off the pump. Other methods can be used, such as dc pulsing of the diodes, increasing or decreasing the pump outside of the region of oscillation, or frequency modulating the pump. (The method of dc pulsing was actually used extensively in microwave PLO experiments.)

Fig. 19 shows a plot of the regions of unconditional switching for various pump frequencies, with a fixed tuning of the PLO. It is to be expected that near the natural frequency of the circuit there is no switching.

The unconditional switching corresponds to the symmetrical triggering of a conventional flip-flop, while forced switching acts as asymmetrical triggering. Unconditional switching is a useful method and was incorporated into a number of PLO systems.

"Tri-Stable" Operation

The characteristic curve of the PLO usually shows a region in which there are three stable states (see Fig. 4); *i.e.*, the device can either oscillate in phase A, in phase B, or not oscillate at all. This region of tri-stability can be used in logical applications, as described below.

If the device is at rest (pump off) and the pump voltage is suddenly applied to put the PLO into the tri-stable region of operation, it will not start to oscillate by itself. But if a locking voltage of either phase and sufficient amplitude is applied, the device will start to oscillate in the phase of the applied signal. A definite threshold is observed: the input must reach a minimum amplitude before the device will start to oscillate (triggers). This makes coincidence switching possible by using linear addition of signals. For instance, we can make the signals (inputs) of such an amplitude that one input alone will not cause triggering, but two of them will.

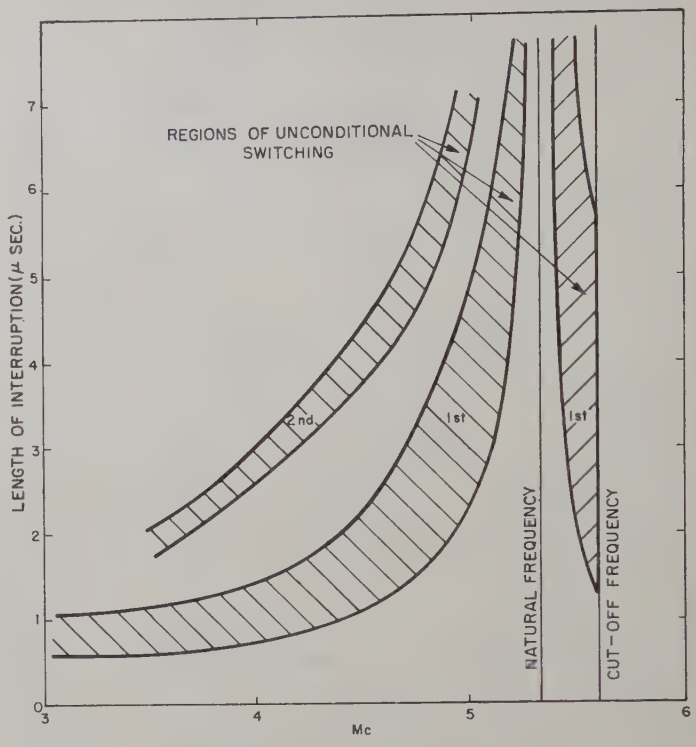


Fig. 19—Unconditional switching curves.

The advantage of using the tri-stable region for logical circuitry lies in the fact that, for special circuits, a whole long array of units—a whole long chain of logic—can be performed by a single pump energization. Once the first units are triggered by the inputs, they trigger the next tri-stable units, etc., until the last ones trigger the output units, in a kind of chain reaction. Thus the direction of information propagation in these special cases can be determined by the combinations of the inputs and not by pump modulation, as in the conventional system. This can be especially useful in some applications, such as carry propagation in binary adders, etc.

Applications

The various applications of PLO in logical circuits were discussed in the last section where the different modes of operation of the device were described. To illustrate the versatility, as well as the shortcomings, of the device this section will describe a random-access memory as a typical example. As the PLO is capable of operating at kilomegacycle frequencies, the possibility of using it to construct a random-access memory with millimicrosecond access time is especially enticing.

Writing into a PLO memory is a simple matter. The forced switching mode of operation can be readily used. The switching threshold of the device allows selective switching by coincidence of inputs. The switching is fast in the forced switching mode.

Selectively reading out one bit of stored information from a PLO memory matrix is not a very easy matter. Two of the more promising approaches will be discussed here.

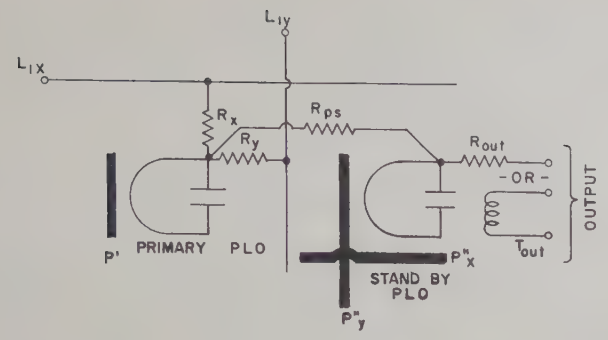


Fig. 20—Stand-by memory system. Circuit for one memory element.

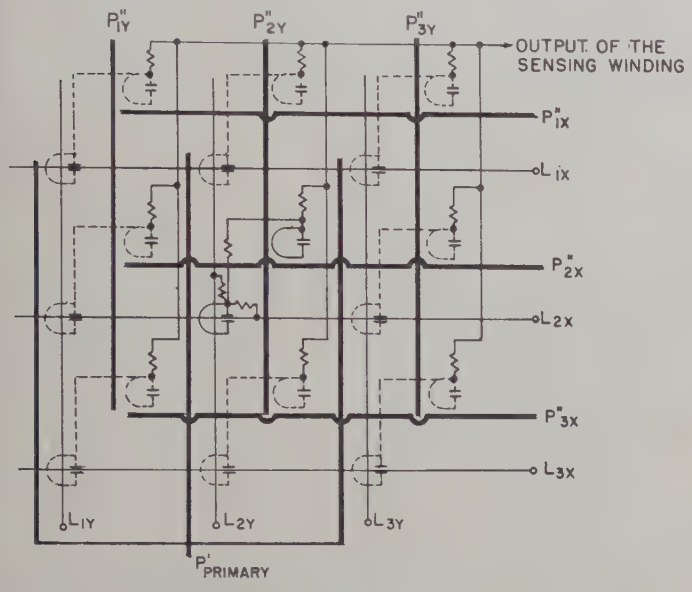


Fig. 21—Digit-plane of the stand-by memory scheme. (Note: sensing winding not shown.)

A straightforward approach to achieving read-out from a PLO memory is to use one PLO unit to store the information, and another to perform the logic of selecting the unit and sensing the information stored in it. A way to accomplish this is shown in Figs. 20 and 21, where two PLO's per bit are used. In this system (called the stand-by system) one of the units is constantly energized and it is here that information is stored. The other unit (the stand-by unit) is normally not energized. During read-out the stand-by unit is selectively energized by the coincidence of two pumps, each of half normal amplitude, in X-Y coincidence. Because of the coupling between the two units, the selected unit will start to oscillate in the same phase as the corresponding storage unit. Since only the selected stand-by unit is oscillating, its output can readily be sensed. Sensing can be accomplished either by mixing through resistors (R_{out} in Fig. 20) or by transformers (T_{out}) connected in series. In a microwave system the output would be transmitted through loosely coupled antenna pick-ups. As many outputs as signal-to-noise ratio will permit can be connected to a single sensing amplifier, which can also be a PLO.

An alternate approach to the PLO Random-Access Memory is shown in Figs. 22 and 23. In this method 2 units per bit are also used, but here both of them are constantly energized. Normally they are oscillating in such a way that their outputs on the sensing winding are 180° out of phase, and thus cancel each other. During read-out the information in one of the units of the selected bit is changed to the reference (B) phase, so that an output results if that unit was originally in the A phase; *i.e.*, if the unit was storing a 1. If the unit was storing a 0 no output would result, since no switching would occur and the two units would still be cancelling each other. After sensing, the disturbed unit has to be brought back to its original condition, or to be restored.

Various organizational schemes are possible, in which the write, read, and restore operations are accomplished in different ways.

In the first place we could have a rewrite loop, where the disturbed unit is restored back to its A phase, if the output indicates that the unit was originally in A phase.

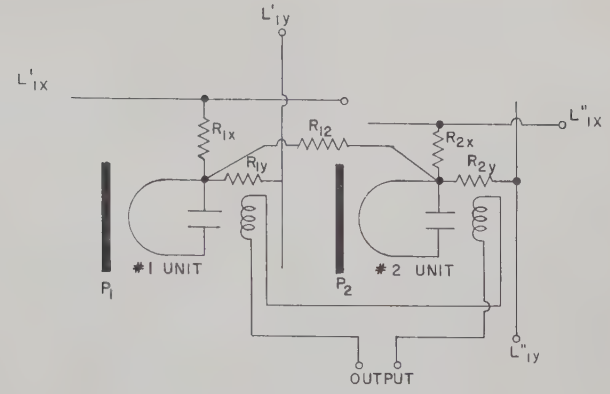


Fig. 22—Cancellation memory system. General circuit for one memory element.

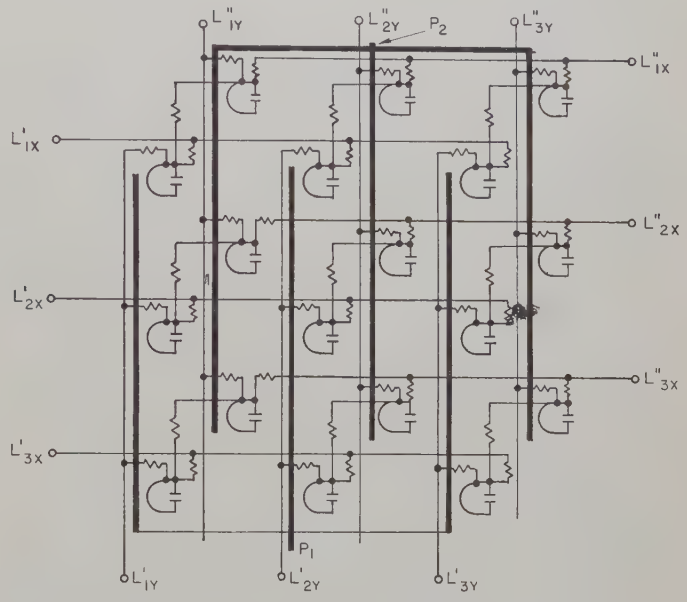


Fig. 23—Digit-plane of the nondestructive cancellation-type memory. [Note: the sensing winding (not shown) threads the digit plane to detect the difference in outputs of the two PLO's for each bit-location.]

To avoid the rewrite loop we can couple the two units (in each bit) and, after the read operation, transfer the information from the primary to the secondary units. This can be done by unconditional interruption of the pump of the secondary units, so that they lock-in to the primary units.

Care must be taken to assure proper relative phasing of the two units; in the normal state the outputs must cancel each other out in the output loop, but be in phase in the restore-coupling loop. Therefore one inversion is necessary, in either one loop or the other.

The restore-coupling loop can also be used to facilitate the write operation; information need only be written into one set of units and then transferred to the other set by using the restore set-up.

The complete read-write cycle would go as follows (Fig. 23):

Write on No. 1 units by forced switching (use either L_x' and L_y' or L_x' and P_1). Transfer from units No. 1 to units No. 2 by modulation of P_2 (by phase initiation method). Read by reversing the phase of No. 2 unit by forced switching (conditional)—either by forced coincidence of L_x'' and L_y'' or L_x'' and P_2. Restore by interruption of P_2, (as in write operation).

If read came before write the same cycle could be used to restore and to complete transfer from units No. 1 to No. 2 for write.

Many variations on this basic cycle are possible.

Conclusions

The various applications and modes of operation of the PLO, which were presented in this paper, were developed during an investigation into the behavior of the device. Results obtained thus far indicate that:

1) the device is capable of performing the basic functions of detection, amplification, and storage of digital information at kilomegapulse clock rate;
2) complete logic and memory systems using only PLO's and linear passive coupling components are feasible;
3) the many possible modes of operation make the PLO a very versatile device. The device is relatively insensitive to reasonable changes in operating conditions and parameter values.

Acknowledgment

The work reported in this paper was sponsored by the Bureau of Ships, Contract Number NObsr 72717. The authors are indebted to Drs. J. A. Rajchman, F. Sterzer, and W. Beam for many helpful suggestions, and to D. Knapschaefer for experimental work at microwave frequencies.

Bibliography

[1] E. Goto, "On the application of parametrically excited nonlinear resonators," *J. Inst. Elec. Commun. Engrs. (Japan)*, vol. 38, pp. 770–775; October, 1955.
[2] J. von Neumann, "Nonlinear capacitance or inductance switching, amplifying, and memory organs," U. S. Patent No. 2,815,488, assigned to IBM; December 3, 1957.
[3] F. Sterzer and W. R. Beam, "Parametric Subharmonic Oscillators," paper presented at the Solid-State Circuits Conf., Philadelphia, Pa.; February, 1959.
[4] Z. Kiyasu, K. Husimi, K. Kataoka, and K. Yamanaka, "Parametric excitation using barrier capacitance of semiconductor," *J. Inst. Elec. Commun. Engrs. (Japan)*, vol. 40, pp. 162–169; February, 1957.
[5] H. Takahashi, "The parametron," *J. Inst. Elec. Commun. Engrs. (Japan)*, vol. 39, pp. 586–590; June, 1956.
[6] J. J. Stocker, "Nonlinear Vibrations in Mechanical and Electrical Systems," Interscience Publishers, Inc., New York, N. Y., 1958.
[7] W. J. Cunningham, "Nonlinear Analysis," McGraw-Hill Book Co., Inc., New York, N. Y., pp. 245–280; 1958.
[8] F. Sterzer, "Random number generator using the parametric subharmonic oscillator," *Rev. Sci. Instr.*, vol. 30, pp. 241–243; April, 1959.

Semiconductor Parametric Diodes in Microwave Computers*

J. HILIBRAND†, C. W. MUELLER†, C. F. STOCKER†, AND R. D. GOLD†

Summary—**The parametric subharmonic oscillator operates by energy transfer from the pump frequency to the oscillator frequency through a nonlinear energy storage element—in the present case, the nonlinear capacitance of a semiconductor diode. This paper examines both the requirements on the diode for satisfactory performance in this circuit and the limitations on oscillator performance which arise from the nature of the semiconductor diode.**

The analysis shows that abrupt junction diodes must have a Q of at least four at the oscillation frequency if there is to be any useable energy transfer, and that graded junction diodes must have a Q of six. The time constant governing the rise of the envelope of the subharmonic waveform is a marked function of the stray capacitances; this function is examined in detail. The choice of bias voltage to obtain the fastest possible rise time involves consideration of the stray capacitance, the Q of the available diode, and limitations imposed by excessive pump power requirements. For negligible stray capacitance, it is shown that the subharmonic waveform can rise by a factor e in 1.3 cycles of the subharmonic frequency for an abrupt junction diode, or in 1.9 cycles for a graded junction diode.

The principles involved in the design of the semiconductor diode are examined and the choice of materials, impurity distributions, and fabrication techniques are discussed. A new diode encapsulation intended for direct mounting in microstrip transmission line is described.

Measurement of appropriate diode parameters is vital to diode research. An equivalent circuit characterization in which the parameters may be directly related to the diode structure is used. Several echniques for the measurem ent of these parameters are discussed.

I. Introduction

VARIABLE capacitance semiconductor diodes are being considered for use in the logic, gain, and storage elements of high-speed computers. Computation functions can be performed by a parametric subharmonic oscillator circuit using a variable reactance element. The computation rate desired imposes some important minimum requirements for acceptable semiconductor diode characteristics. At the same time, the use of semiconductor diodes with less than ideal properties will impose limitations on the performance of the subharmonic oscillator. It is this interaction between the requirements on the diode and the limitations which stem from the use of nonideal diodes, together with the meaning of this interaction in terms of diode design, which is the substance of this paper.

The parametric subharmonic oscillator operates by energy transfer from a pump frequency to the oscillator frequency through the medium of a nonlinear reactance. For a pump frequency equal to twice the oscillator frequency, energy transfer will occur to give oscillations in either of two stable phases. It is this bistable phasing that has made the circuit of potential importance.[1] The use of nonlinear inductance was contemplated at first and, indeed, a computer has been built using nonlinear inductances.[2] Recently, it has been recognized that the nonlinear capacitance of a semiconductor diode offers the possibility of using higher oscillator frequencies, giving faster computing rates.[3]

A wide variety of semiconductor diodes has become available in the past decade. Point contact microwave diodes have long been used as detectors up to millimeter wave frequencies,[4] but these units are designed to be nonlinear resistances; the associated nonlinear capacitance characteristics are not exploited. Gold-bonded germanium diodes have served as reliable nonlinear resistors for moderate speed base-band computers, and more recently have been used in variable capacitance applications at microwave frequencies.[5] P-n junction diodes with abrupt transitions[6] or variously graded transitions[7] from p- to n-type have been designed for specific application as variable capacitance diodes.

Semiconductor diodes fall short of an ideal variable capacitance. There are various losses which restrict the maximum frequency and the voltage amplitude range of the circuit in which the diode is applied. In addition, the voltage dependence of the capacitance may not be as pronounced as might be desired; this limits the rate of energy transfer in the circuit. In this paper, some of

* Manuscript received by the PGEC, April 30, 1959. Presented at the ONR Symp. on Microwave Techniques for Computing Systems, Washington, D. C., March 12, 1959. This work was performed under U. S. Navy BuShips contracts.

† RCA Laboratories, Princeton, N. J.

[1] J. von Neumann, "Nonlinear capacitance or inductance switching, amplifying, and memory organs," U. S. Patent No. 2,815,488; December 3, 1957.

[2] E. Goto, "On the application of parametrically excited nonlinear resonators," *Denki Tsushin Gakai-shi*, vol. 38, pp. 770–775; October, 1955. Also, S. Muroga and K. Takashima, "The parametron digital computer MUSASINO-1," this issue, p. 308.

[3] Z. Kiyasu, K. Fushimi, K. Yamanaka, and K. Kataoka, "Parametric excitation using the barrier capacitance of a semiconductor," *J. Inst. Elec. Commun. Engrs. (Japan)*, vol. 40, pp. 162–169; February, 1957. I. S. Onyshkevych, W. F. Kosonocky, and A. W. Lo, "Parametric phase-locked oscillator—characteristics and applications to digital systems," this issue, p. 277. W. R. Beam, D. J. Blattner, and F. Sterzer, "Microwave carrier techniques for high speed digital computing," presented at the Symp. on Microwave Techniques for Computing Systems, Dept. of Interior, Washington, D. C.; March 12, 1959.

[4] G. C. Messenger, "New concepts in microwave mixer diodes," PROC. IRE, vol. 46, pp. 1116–1121; June, 1958.

[5] K. L. Kotzebue, "A Semiconductor-Diode Parametric Amplifier at Microwave Frequencies," Stanford Electronic Labs., Stanford University, Stanford, Calif., Tech. Rept. No. 49; November, 1958.

[6] L. J. Giacolletto and J. O'Connell, "A variable capacitance germanium junction diode for UHF," *RCA Rev.*, vol. 17, pp. 68–85; March, 1956.

[7] A. Uhlir, Jr., "The potential of semiconductor diodes in high-frequency communications," PROC. IRE, vol. 46, pp. 1099–1115; June, 1958.

these limitations on circuit performance which stem directly from the properties of the diode will be examined. This will be accomplished by first obtaining an approximate linear circuit in which the energy transfer process is related to the diode parameters, and then analyzing the properties of this circuit. Having established circuit performance in terms of the diode, the problems of diode design will be considered—as applied to both the semiconductor diode structure itself and to the diode encapsulation. Various measurement techniques for establishing the pertinent diode parameters will be described, together with some preliminary results of these measurements.

II. The Semiconductor Diode as a Circuit Element

A semiconductor diode can be represented by the equivalent circuit shown in Fig. 1. The small-signal capacitance $C(V)$ and conductance $G(V)$ are functions of the bias voltage:[8]

$$C(V) = K(\phi - V)^{-n} \tag{1}$$

$$G(V) = \frac{q}{kT} I_0 e^{qV/kT} \tag{2}$$

as shown in Fig. 2(a) and (b). The voltage ϕ is the junction contact potential, q is the electron charge, k is the Boltzmann constant, and T is the temperature in degrees Kelvin. K is a constant depending on material and structural properties. The exponent $n = \frac{1}{2}$ for abrupt junction diodes and $\frac{1}{3}$ for linearly graded junction diodes. The series resistance r_s is usually independent of diode voltage.[9]

At high frequencies, and for negative or slightly positive biases, the capacitance $C(V)$ shunts out the conductance $G(V)$, and r_s is the only important lossy element in the circuit. A measure of the ratio of the average energy stored in the capacitor to the energy dissipated in the resistor r_s per cycle is the quality factor

$$Q = \frac{1}{\omega C r_s} \tag{3}$$

where ω is the subharmonic frequency determined by computer system requirements. Since C is here a function of voltage, the above energy-power relation will be true only for small fluctuations around an operating point. This series-loss quality factor, Q_s, will be defined by (3) at the bias voltage V_0. If this capacitor is to be used in a tuned circuit, it is clearly desirable that Q_s be appreciably greater than unity so that the energy circulating between inductance and capacitance will not be dissipated too rapidly.

When the diode is forward-biased, the conductance $G(V)$ increases rapidly with voltage, and losses in $G(V)$

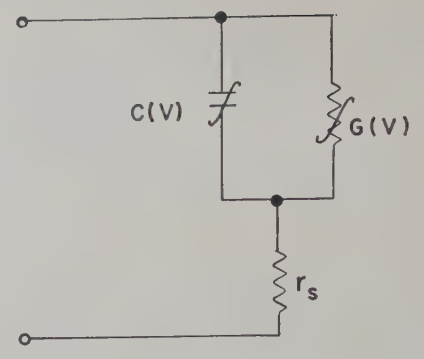


Fig. 1—Equivalent circuit for a semiconductor diode.

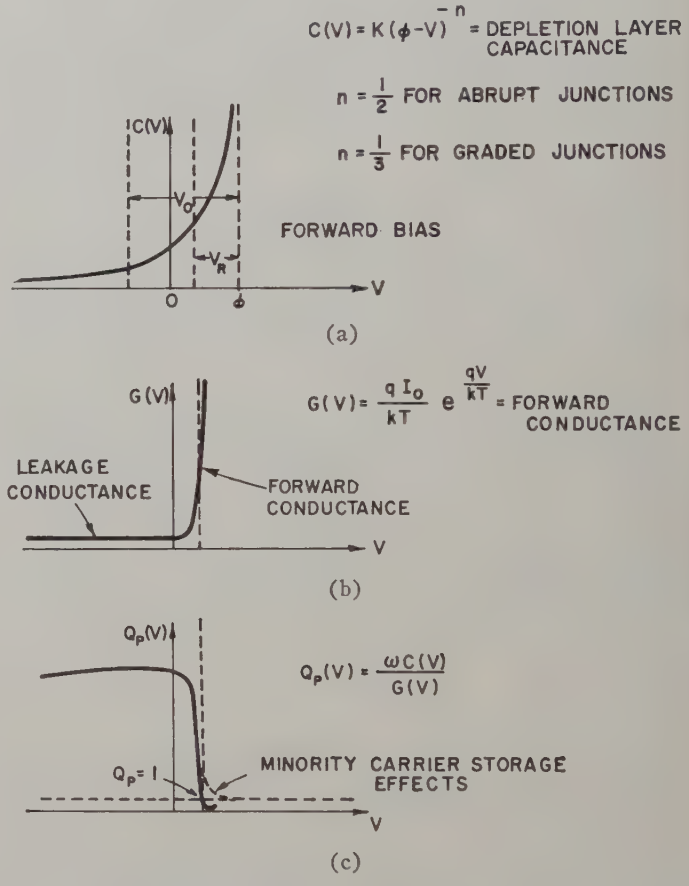


Fig. 2—The voltage-variable diode parameters.

may become serious enough to limit the over-all Q of the diode. For a conductance in parallel with a capacitance, the quality factor Q_p is written as

$$Q_p = \frac{\omega C(V)}{G(V)} . \tag{4}$$

As in the case of the series-loss quality factor, Q_p is defined on a small signal basis and is a function of the instantaneous diode voltage. The value of Q_p determines the over-all diode loss factor for forward bias voltages as in Fig. 2(c). The voltage at which Q_p falls to unity[10] is used to define the maximum allowed instantaneous for-

[8] W. Shockley, "The theory of p-n junctions in semiconductors and p-n junction transistors," *Bell Sys. Tech. J.*, vol. 28, pp. 435–489; July, 1949. Minority carrier storage effects do not appear in the equations above. They will be considered later in this paper.

[9] This series resistance varies only slightly for reverse bias, although some anomalous resistance variations have been noted.

[10] When minority carrier storage effects are taken into account, it will be shown that $Q_p = 2$ determines the limiting voltage.

ward voltage on the diode. It is convenient to define the difference between this maximum forward voltage and the contact potential as V_R [see Fig. 2(a)]. Since Q_p usually falls off rapidly near this voltage $\phi - V_R$, if the diode is biased so that the instantaneous diode voltage is in this region for only a small part of each cycle, the time-averaged losses in the parallel conductance will be small; the major contribution to the diode losses will stem from the resistor r_s. Operation with such a limited voltage swing will be assumed in the following sections and Q_s will be used to specify the diode quality factor.

The frequency at which Q_s goes to unity is commonly referred to as the cutoff frequency f_{co}. It will be noted that this parameter, f_{co}, is defined in this paper at the operating point rather than for maximum negative bias as is sometimes done in diode specifications. This results in a value which is more meaningful for circuit performance.

III. An Approximate Linear Model and its Properties

A. A Linear Model for the Diode

The basic lumped constant circuit used for the parametric subharmonic oscillator is shown in Fig. 3. A power source at frequency 2ω is applied to a tuned circuit consisting of the diode, various stray capacitances, and an inductance chosen to tune the circuit to frequency ω. The voltage across the capacitor will consist of various harmonics and cross-products between these frequencies. The voltages at frequencies ω and 2ω might be expected to be the only ones of any appreciable amplitude since the first is the frequency for which the circuit is tuned and the second is the frequency at which the circuit is driven. The voltage across the diode will, then, be of the form

$$v(t) = V_s \sin(\omega t + \theta) + V_p \sin 2\omega t \tag{5}$$

where V_s is the subharmonic voltage amplitude, V_p is the pump voltage amplitude, and θ is, as yet, an undetermined phase angle.

The capacitance variation with voltage can be represented by a Taylor series expansion around the operating point $V = \phi - V_0$ as in Fig. 2(a):

$$\begin{aligned} C(V) &= K(\phi - V + v)^{-n} = K(V_0 + v)^{-n} \\ &= (KV_0^{-n}) - n(KV_0^{-n})\frac{v}{V_0} \\ &\quad + \frac{n(n+1)}{2!}(KV_0^{-n})\left(\frac{v}{V_0}\right)^2 - + \cdots \\ &= C_0 - nC_0\frac{v}{V_0} + \frac{n(n+1)}{2!}C_0\left(\frac{v}{V_0}\right)^2 - + \cdots. \end{aligned} \tag{6}$$

It can be shown[11] that the first two terms in this series provide a good approximation to the capacitance function.

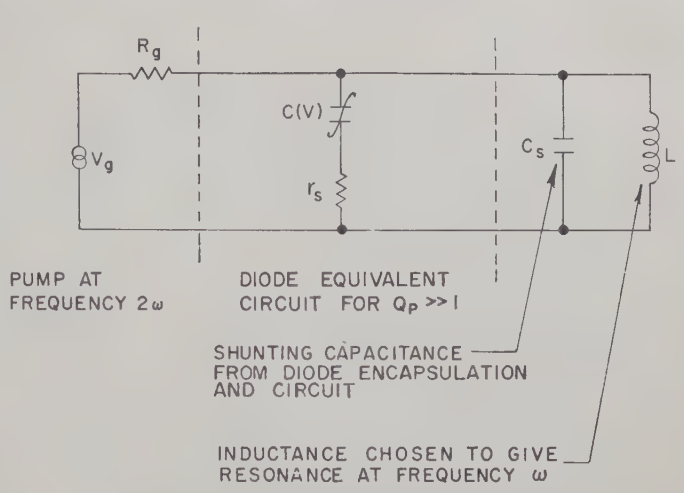


Fig. 3—The parametric subharmonic oscillator.

An equivalent linear circuit for the capacitor at frequency ω will now be established. Such a circuit is found by examining the ratio of the voltage at frequency ω to the current flowing at frequency ω into the capacitance terminals. The total current is given by $i = C(V)dv/dt$. The terms in the expression for the current at frequency ω are

$$\begin{aligned} i_\omega &= (KV_0^{-n})\omega V_s \cos(\omega t + \theta) \\ &\quad + \frac{n(KV_0^{-n})}{2}\frac{V_p}{V_0}\omega V_s \sin(\omega t - \theta). \end{aligned} \tag{7}$$

The first term is indicative of the presence of a capacitance of magnitude $KV_0^{-n} = C_0$. The second term relates to energy transfer between the pump and oscillation frequencies and is dependent on the phase angle θ. When θ is 0 or 180°, it is a positive conductance or a loss of oscillation frequency energy to the pump circuit. When θ is 45° or 135°, it is a pure reactance, indicating an interchange of energy between the pump and subharmonic frequencies but no net power transfer. For θ equal to $\pm 90°$, the second term is a negative conductance. Here, power is being fed from the pump frequency to the subharmonic frequency. The subharmonic oscillations will clearly tend to build up and saturate in one of these two phases. For the choice of $\theta = \pm 90°$, the equivalent circuit at frequency ω consists of a negative conductance,

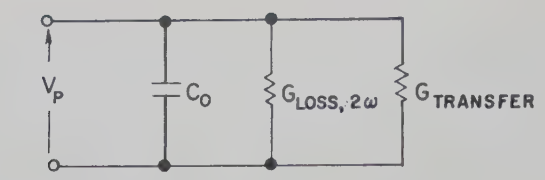
$$G_- = -\frac{n\omega}{2}(KV_0^{-n})\frac{V_p}{V_0} \tag{8}$$

and a capacitance, C_0.

An equivalent circuit at frequency 2ω can also be established. The terms in the expression for the current at this frequency (assuming $\theta = \pm 90°$) are:

$$i_\omega = (KV_0^{-n})2\omega V_p \cos 2\omega t + \frac{n\omega}{2}(KV_0^{-n})\frac{V_s^2}{V_0}\sin 2\omega t. \tag{9}$$

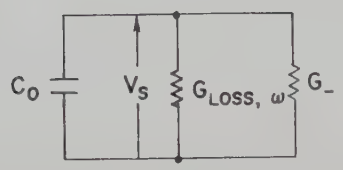
[11] J. Hilibrand and W. R. Beam, "Semiconductor diodes in parametric subharmonic oscillators," *RCA Rev.*, vol. 20, pp. 229–253; June, 1959.



$$G_{LOSS,2\omega} = \frac{4\omega C_0}{Q_S}$$

$$G_{TRANSFER} = \frac{n\omega C_0}{2} \frac{V_S^2}{V_0 V_P}$$

(a)



$$G_{LOSS,\omega} = \frac{\omega C_0}{Q_S}$$

$$G_- = -\frac{n\omega C_0}{2} \frac{V_P}{V_0}$$

(b)

Fig. 4—Linearized form of diode equivalent circuits at frequencies ω and 2ω; (a) diode equivalent circuit at pump frequency 2ω; (b) diode equivalent circuit at subharmonic frequency ω.

The first term is again a capacitance of value $KV_0^{-n} = C_0$. The second term is a positive conductance of value:

$$G_{transfer} = \frac{n\omega}{2}(KV_0^{-n})\frac{V_s^2}{V_0 V_p} \tag{10}$$

and represents the net power transfer from the pump to the oscillator frequency. The energy at frequency 2ω entering $G_{transfer}$ appears at frequency ω in $G-$.

The losses in the diode series resistance can be represented in this admittance model (for $Q_s > 1$) by the equivalent conductance at each frequency:

$$G_{loss,\omega} = (\omega C_0)^2 r_s = \frac{\omega C_0}{Q_s},$$

$$G_{loss,2\omega} = (2\omega C_0)^2 r_s = \frac{4\omega C_0}{Q_s}.$$

The final pair of diode equivalent circuits appears in Fig. 4. Using these circuits, the energy transfer process can be examined in detail.

B. The Optimum Negative Conductance and the Minimum Diode Quality

The energy transfer process can be optimized by suitable choice of bias voltage. Also, the net negative conductance in the presence of diode losses can be examined and leads to a requirement on minimum diode quality for the subharmonic oscillator application.

The relation for negative conductance in (8) is a function of the bias and pump voltages. That is, G_- is proportional to the pump voltage V_p and to the slope of the C-V characteristic at voltage V_0, $n(KV_0^{-n})/V_0$. Clearly, it is desirable to increase both of these. However, the pump and bias voltages are related by the requirement that forward conduction losses in $G(V)$ are to be limited. If the voltage $\phi - V_R$ in Fig. 2 is taken as the maximum forward voltage, and if the subharmonic voltage V_s is negligibly small compared to V_p (as it normally is), the pump voltage is

$$V_p = V_0 - V_R.$$

Using this relation, we find

$$G_- = -\frac{n\omega}{2}(KV_0^{-n})\left(1 - \frac{V_R}{V_0}\right). \tag{11}$$

The bias voltage to maximize G_- is easily found to be

$$V_0 = \frac{1+n}{n} V_R \tag{12}$$

and the maximum negative conductance is

$$G_{-max} = -\frac{n\omega}{2}\frac{1}{n+1}(KV_0^{-n}) = -\frac{n\omega C_0}{2(n+1)}. \tag{13}$$

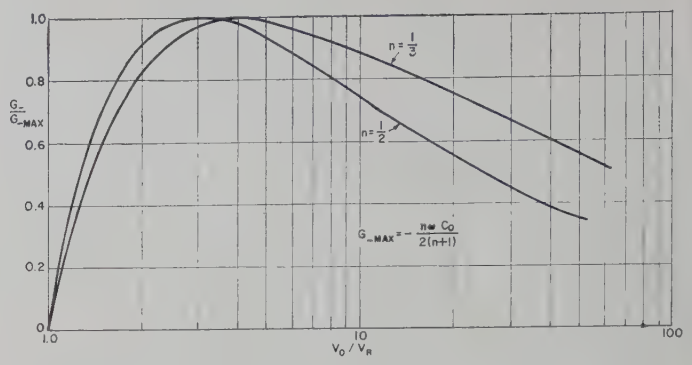


Fig. 5—The effect of bias voltage on the negative conductance.

The variation of this negative conductance with bias voltage is shown in Fig. 5 in terms of G_-/G_{-max} for $n=\frac{1}{2}$ and $n=\frac{1}{3}$, two values of interest. The optimum is not a very sharp one and it is clear that one can vary the bias voltage V_0 over a 3:1 range losing less than 10 per cent of the maximum negative conductance.

The maximum negative conductance is of importance in situations where the shunting capacitance in the subharmonic oscillator circuit is large compared to the diode capacitance itself. For such high shunt capacitances, it will be shown that the optimum oscillator rise time is obtained using the bias for maximum negative conductance. This bias voltage will also be referred to in considerations of pump power dissipation.

The minimum diode quality factor is determined by the maximum acceptable loss of energy at the subharmonic frequency in the series resistance r_s. When the negative conductance, G_-, and the positive conductance $\omega C_0/Q_s$ are equal, there will be zero net energy transfer to the subharmonic frequency. Equating these two con-

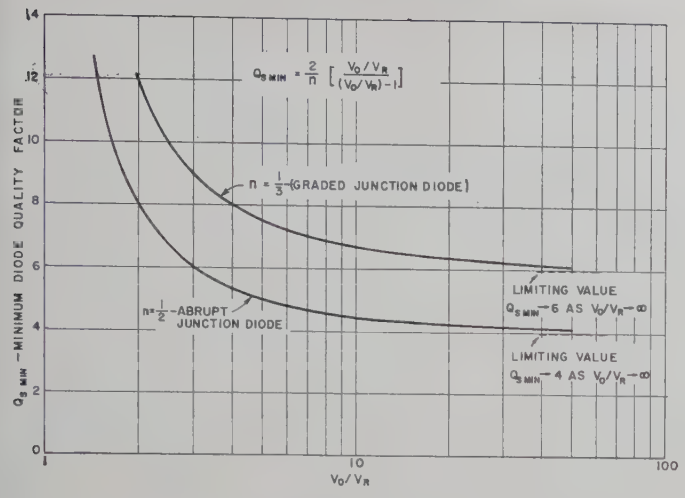


Fig. 6—Effect of bias voltage on the minimum diode quality factor.

ductances specifies the minimum acceptable diode quality factor at the chosen bias voltage, $Q_{s\,min}$.

$$Q_{s\,min} = \frac{2}{n}\left(\frac{V_0/V_R}{V_0/V_R - 1}\right). \tag{14}$$

$Q_{s\,min}$ is shown as a function of bias voltage in Fig. 6.

It will be noted that for abrupt junction diodes, the limiting value of the minimum quality factor is four and for graded junction diodes, the minimum quality factor is six. The choice of bias voltage is limited primarily by pump power dissipation, as will be shown in section III-D so that these minimum values may be optimistic.

C. Rise Time in the Subharmonic Oscillator

When this circuit is applied in computers, the rise times attainable at the subharmonic frequency will limit the computation rate. This can be evaluated analytically. In this section, the problem of choosing bias voltage to optimize the subharmonic voltage rise time will be considered. It will be demonstrated that the bias voltage for minimum rise time corresponds to the bias voltage for maximum negative conductance only where the shunt capacitances are far greater than the diode capacitance.

Consider the circuit of Fig. 7 which is the tuned equivalent circuit at the subharmonic frequency. In this figure, $C_T = C_s + C_0$ is the sum of various shunting capacitances and the diode capacitance. The rise time of the envelope of the subharmonic waveform (the time for the envelope to increase by a factor e) is

$$\tau = \frac{2C_T}{G_{NET}}.$$

The basic quantity of interest is the number of cycles of the subharmonic in one such time constant interval, which is equal to τf. Substituting for the quantities involved in the expression for the time constant and assuming diode losses are negligible, it can be shown[11] that

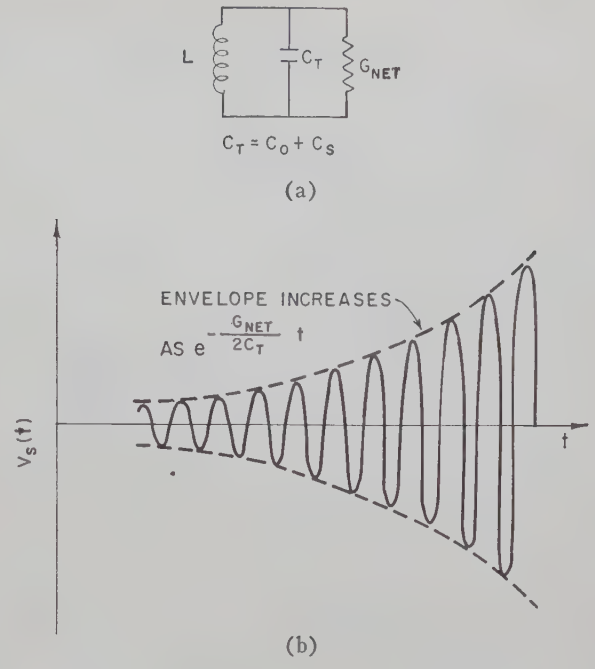


Fig. 7—Oscillator equivalent circuit response at the subharmonic frequency; (a) oscillator equivalent circuit; (b) equivalent circuit response at frequency ω for G_{NET} negative.

$$\tau f = \frac{2}{\pi n}\left(\frac{V_0/V_R}{V_0/V_R - 1}\right)\left[1 + \frac{C_s}{C_R}\left(\frac{V_0}{V_R}\right)^n\right] \tag{15}$$

where $C_R = K V_R^{-n}$ is the diode capacitance at the maximum forward voltage. This function, τf, is shown in Fig. 8 for an abrupt junction diode and in Fig. 9 for a graded junction diode. It will be noted that stray capacitances can have an appreciable effect on the rise time. Also, the bias point for optimum rise time corresponds to a greater reverse bias than that for maximum negative conductance. The abrupt junction diode can rise by a factor e in 1.27 cycles of the subharmonic while the graded junction diode will require 1.91 cycles of the subharmonic. These represent ultimate values and cannot be achieved in practical circuits where there are restrictions on the bias voltage due to considerations of avalanche breakdown and pump-power limitations.

As would be expected, the effect of diode losses is to deteriorate the rise time even further, since the net negative conductance, which is the only source of subharmonic power, decreases. Another degradation of rise time can stem from lead inductance, if it is appreciable.

D. Power Dissipation in the Subharmonic Oscillator

It is necessary that the pump power requirements be reasonable if the subharmonic oscillator is to be a practical element in a computer. The pump power is dissipated primarily in the conductance $G_{loss,\,2\omega}$ of Fig. 4 if the subharmonic amplitude is less than the pump amplitude ($V_s < V_p$) as is usually the case.

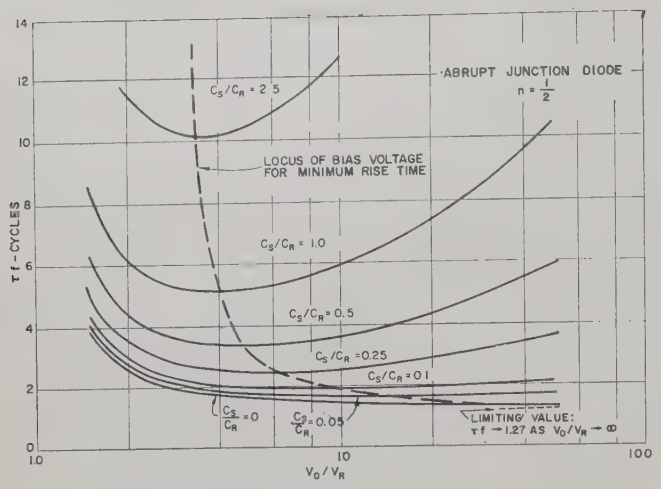


Fig. 8—Subharmonic oscillator rise time ($n=\frac{1}{2}$).

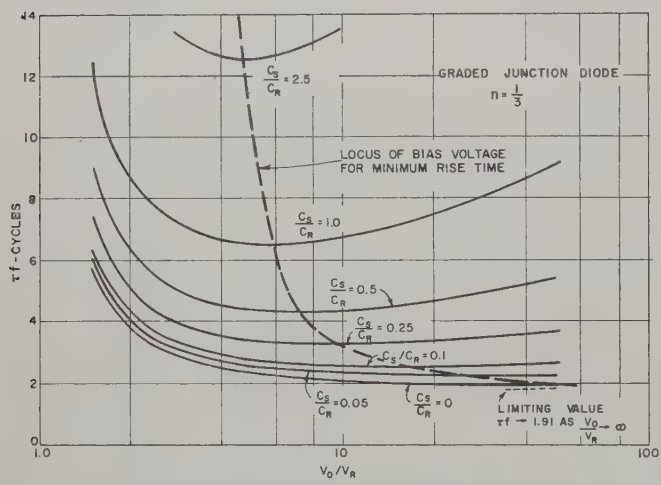


Fig. 9—Subharmonic oscillator rise time ($n=\frac{1}{3}$).

The power lost is

$$P = \frac{V_p^2}{2}\,\frac{4\omega C_0}{Q_s} = \frac{2\omega C_0 V_p^2}{Q_s}. \tag{16}$$

If the bias voltage is chosen to yield the maximum negative conductance, and the optimum pump voltage for that bias is used, $V_p = V_R/n$. Then

$$P_0 = \frac{2\omega C_0 V_R^2}{n^2 Q_s}. \tag{17}$$

The effect of choosing larger values of bias voltage is to give a very marked increase in pump power requirements:

$$\frac{P}{P_0} = n^2\left(\frac{V_0}{V_R} - 1\right)^2. \tag{18}$$

This function is plotted in Fig. 10. A typical value for P_0 is

$$P = \frac{2\omega C_0 V_R^2}{n^2 Q_s} = 3 \text{ mw};$$

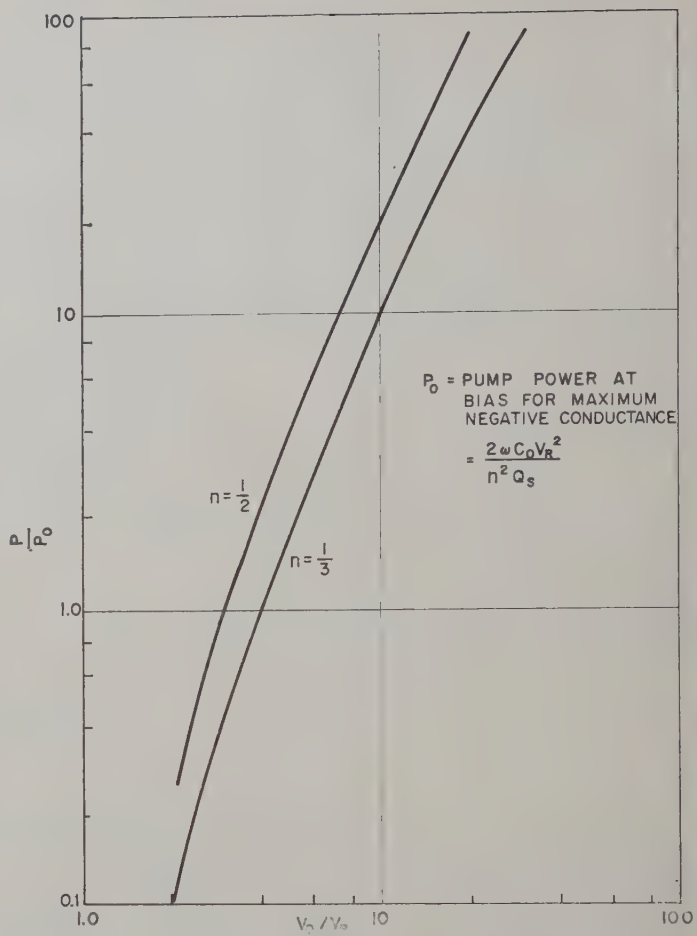


Fig. 10—Pump power required as a function of bias voltage.

if

$$V_R = 0.4 \text{ v},$$

$$C_0 = 1\ \mu\mu\text{f},$$

$$\omega = 2\pi \times 10^{10} \text{ rps},$$

$$n = 1/2,$$

$$Q_s = 25.$$

This certainly is an acceptable dissipation. If, to achieve a more rapid rise time, V_0/V_R is chosen to be 10, $P = 60$ mw which is still acceptable, though perhaps a trifle high, for a large system of such elements in a small volume.

This examination of the pump-power limitations emphasizes that the choice of bias for optimum rise time is subject to pump-power dissipation limitations and that the value of V_R should be as small as possible.

IV. Design of the Semiconductor Diode

A. Factors in Diode Performance

The circuit requirements on semiconductor diodes for use in parametric subharmonic oscillators have been

established. These requirements will be summarized below and their significance examined in terms of semiconductor diode design.

The important diode properties are the ratio of shunting capacitance to junction capacitance at maximum voltage C_s/C_R, the diode quality factor at the bias point Q_s, the exponent of the C-V characteristic n, and the maximum forward voltage taken with respect to contact potential V_R.

The first factor, the ratio C_s/C_R, is determined almost entirely by the diode encapsulation and will be discussed when the encapsulation is described. The diode quality factor Q_s is perhaps the most important diode property. Since the minimum diode capacitance is determined, in a practical sense, by the diode encapsulation capacitance, a high-quality factor requires that r_s be made as small as possible. However, series resistance in the oscillator circuit itself imposes a lower practical limit on this parameter as will also be discussed in section IV-E. The exponent involved in the C-V characteristic, n, plays an important role in determining the minimum acceptable diode quality factor and the oscillator rise time. Both these parameters call for a sensitive dependence of C on V; *i.e.*, a large exponent, n. A compromise between increasing cutoff frequency and increasing values of n will be considered in a later section. The voltage V_R will be shown to be minimized by heavy doping, thus permitting operation further in the forward region without excessive losses.

Other diode parameters, of somewhat less importance to circuit performance, are breakdown voltage of the diode, and the lead inductance. So long as breakdown voltage is large enough and the lead inductance is small enough, these parameters have no effect on the circuit.

B. Illustrative Design of a Planar Abrupt Junction Diode

The design of a semiconductor diode is governed by the competition and interaction between restrictions imposed by the circuit, the semiconductor materials, and the processing and encapsulating technology. This section will be used to illustrate the compromises involved in choosing diode dimensions, doping, and materials. A practical diode capable of achieving ultimate performance will not result from this illustrative example, but the direction of modifications which will lead to the ultimate performance will be indicated.

The cutoff frequency f_{co}, the frequency at which $Q_s = 1$, is independent of diode area since the capacitance is proportional and the series resistance inversely proportional to area. The values of r_s, C_0 and f_{co} are given for the abrupt junction diode model in which one region is much more heavily doped than the other:[12]

$$C_0 = A\sqrt{\frac{N_d q \epsilon}{2V_0}}$$

$$r_s = \frac{t}{\mu_n N_d q A}$$

$$f_{co} = \frac{1}{2\pi r_s C_0} = \frac{\sqrt{2q}}{2\pi}\cdot\sqrt{\frac{\mu_n^2 N_d}{\epsilon}}\cdot\sqrt{V_0}\cdot\frac{1}{t} \qquad (19)$$

where

q = electronic charge,
μ_n = electron mobility in the semiconductor,
N_d = donor impurity density in lightly doped n-type region,
ϵ = dielectric permittivity in the semiconductor,
V_0 = bias voltage measured from the contact potential as in Fig. 2,
A = junction area, and
t = thickness of the n-type region.

In the expression for cutoff frequency, the parameters have been grouped to show that the second term, $\sqrt{(\mu_n^2 N_d)/\epsilon}$, represents those parameters which are functions of the material, the third term, $\sqrt{V_0}$, is a circuit parameter, and $1/t$ is a structural parameter. The first term is a constant.

The material parameters are subject to optimization for a given semiconductor material. Mobility decreases with increasing doping density and the product $\mu^2 N_d$ reaches a maximum for some semiconductor materials. In n-type germanium, this maximum occurs near $N_d = 10^{18}/\text{cm}^3$ and is broad. It is usually found desirable to use a smaller doping than this, resulting in a higher impedence level in the diode and an acceptable breakdown voltage. The choice of semiconductor materials for high-frequency diodes has been previously discussed.[13] A comparison of cutoff frequencies for germanium, silicon, and gallium arsenide shows that n-type GaAs seems to be a factor of 1.2 better than n-type germanium and that silicon is considerably poorer than either.

The circuit parameter in f_{co}, $\sqrt{V_0}$, is usually determined by the maximum acceptable pump-power dissipation. For some diodes, a low breakdown voltage provides a further constraint on the value of V_0.

The structure parameter in cutoff frequency $1/t$ is determined largely by technological considerations such as the minimum wafer thickness possible without excessive breakage and the maximum controllable alloying depth for small junctions and for ohmic contacts.

A typical choice of parameter values for n-type germanium is $t = 10^{-3}$ cm, $N_d = 10^{17}/\text{cm}^3$ and $V_0 = 1$ v. The corresponding cutoff frequency is 60 kmc. An acceptable impedance level is selected by choice of the junction

[12] The lightly doped region is chosen n-type because of the higher mobility of electrons. It is assumed that the series resistance stems entirely from the n-type region.

[13] D. A. Jenny, "A gallium arsenide microwave diode," Proc. IRE, vol. 46, pp. 717–722; April, 1958.

area. For an area 2 mils $\times 2$ mils, the bulk series resistance r_s is 1 ohm, and the capacitance is $C_0 = 2.7$ $\mu\mu$f.

It is appropriate to indicate here that the bulk series resistance described above is frequently augmented by appreciable resistance in the ohmic contacts to the semiconductor. Furthermore, surface layers can increase the junction capacitance beyond its nominal value.

The value of V_R is an important diode parameter. In diodes used at high frequencies, where the lifetime of injected minority carriers is long compared to the reciprocal of the frequency, storage effects will govern the diode forward admittance. The effect of minority carrier storage is to give a diode admittance having unity quality factor for large forward biases. The definition of the voltage $\phi - V_R$ must be modified since $Q_p = 1$ is no longer a uniquely defined point. An appropriate value of quality factor to be used is $Q_p = 2$. This corresponds to the voltage at which the quality factor falls off precipitately from the Q_s value, as shown in Fig. 2.

When $Q_p = 2$, the diode forward conductance with storage, the storage susceptance and the depletion layer susceptance are all equal. The forward conductance with storage is[8]

$$G_s = G_0\sqrt{\frac{\omega\tau_p}{2}} = \sqrt{\frac{\omega\tau_p}{2}}\,\frac{q\mu_p}{L_p}\,\frac{n_i^2}{N_d}\,A e^{qV/kT} \quad (20)$$

where

G_0 = forward conductance at low frequencies,
τ_p = minority carrier lifetime in the n-type region,
μ_p = hole mobility in the n-type region,
n_i = intrinsic carrier density,
L_p = hole diffusion length in the n-type region.

The depletion layer capacitance, C_T, is

$$C_T = \sqrt{\frac{N_d q\epsilon}{2(\phi - V)}}\,A.$$

The voltage V_R is defined by the requirement that ωC_T equal G_s, or that[14]

$$1 = \sqrt{\frac{\omega\epsilon}{q\mu_p}}\,\frac{N_d^{5/2}}{(N_c N_v)^{3/2}}\,\frac{e^{qV_R/kT}}{\sqrt{\frac{qV_R}{kT}}},$$

where

N_c = effective density of states in the conduction band and
N_v = effective density of states in the valence band.

For the abrupt junction diode described above, the value of V_R is 0.55 volt. An increase of N_d from 10^{17}/cm^3 to 10^{18}/cm^3 would reduce V_R to 0.4 volt and decrease pump-power dissipation by a factor of 2.

It is notable that the value of V_R is almost independent of the material parameters and would not differ appreciably for various semiconductors doped to the same impurity level. Since the capacitance variation is determined primarily by the maximum forward bias, this indicates that the capacitance variation differs only slightly between semiconductor materials.

C. Improvements Over a Planar Abrupt Junction Diode

In the relations for capacitance and series resistance in (19), the doping level is subject to compromise. Light doping corresponds to low capacitance but high resistance. This difficulty is reduced by using a nonuniform doping distribution. A low impurity density is used in the immediate region of the p-n junction, while a high doping level is used in the bulk to reduce series resistance. Such a nonuniform impurity distribution can be achieved in semiconductor diodes by appropriate use of diffusion techniques. This matching of the doping density to the different functions performed in the two semiconductor regions gives promise of perhaps a tenfold higher cutoff frequency than for the abrupt junction diode previously considered.

For most planar diodes, the area in the resistance relation is the same as the area in the capacitance relation of (19). But for junctions whose radius is smaller than the length of the conducting path (or base thickness t), the flow of carriers spreads out from the junction and the effective area in the resistance relation is much greater than the capacitive area, giving an improvement in the cutoff frequency. It is the operation of this spreading resistance principle in point contact diodes which led to the early achievement of high cutoff frequencies in those diodes. It is important to note, on the other hand, that for junction diodes of minimum practical area, it is quite feasible to make the final wafer thickness so small that a series resistance less than the spreading resistance is achieved.

D. The Diode Encapsulation

It has been pointed out earlier that it is desirable to design a diode encapsulation which will exhibit low capacitance. Several other requirements must be met by the encapsulation. There must be small lead inductance if this parameter is not to degrade oscillator rise time or to impose highly restrictive minimum diode impedance levels. The encapsulation must be designed to fit into a microstrip transmission line without unduly disturbing the line properties. Finally, it must protect the diode from the ambient air for long-term reliability.

A diode encapsulation designed to perform these several functions is shown schematically in Fig. 11. An insulator is sealed between two metal plates, with a metal finger entering from the top and contacting the dot on the wafer. This is to be a hermetically sealed enclosure. The upper and lower tabs may be soldered to the two

[14] Use is made of Fermi-Dirac statistics to obtain this result: if Eg is the width of the forbidden band, $\eta_c^2 = N_v N_c\, e^{-qEg/kT}$, and for nondegenerate dopings, $Eg - \phi = kT/q \ln {}^{Nd}/\sqrt{N_v N_c}$. The relation for V_R follows directly.

conductors of a microstrip transmission line with a low melting point solder. Preliminary versions of this encapsulation had a case capacitance of 0.5 $\mu\mu$f and a lead inductance of 0.6 mμh.

Since the diode encapsulation itself will provide a large part of the unavoidable shunt capacitance, the minimum enclosure capacitance will figure in the choice of diode parameters. The diode encapsulation capacitance does not seem to be easily reducible much below a value of 0.5 $\mu\mu$f. A requirement that C_s/C_R be less than unity imposes a lower limit on the suitable diode junction capacitance of 0.5 $\mu\mu$f.

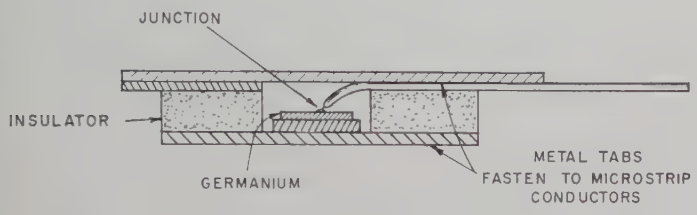


Fig. 11—Cross section of diode.

With the minimum useful junction capacitance determined primarily by the case shunt capacitance, the requirement for a reasonable quality factor at the operating frequency can be satisfied only by a low series resistance. A lower limit for profitable reduction of the diode series resistance is imposed by the various series resistances in the oscillator circuit which, considering skin effects at frequencies such as 10 kmc, is not unimportant. A reasonable limit on the series resistance from this source would be perhaps 0.5 ohm. Using this value of series resistance and the minimum capacitance previously obtained, a Q of 64 at 10 kmc would appear to be a limiting value for practical encapsulated diodes. There is certainly sufficient room for design compromises between this maximum possible Q of 64 and the necessary minimum Q of 4 required for oscillator operation.

E. Practical Semiconductor Diodes

Several approaches to diode fabrication have been pursued, namely, point contacts, alloy junctions, diffused junctions and combinations of these. The compromises dictated by theoretical design and the practical execution are so varied that an *a priori* conclusion regarding the best method of fabrication cannot be formed at present. Consequently, several types of diodes have been fabricated by different processes.

Since noise is not a major consideration and since small capacitances can easily be achieved, point contact diodes were fabricated. Out-diffused n-type germanium wafers gave useful point contact diodes. The very small capacitance of these units made direct measurement of r_s very difficult and, therefore, circuit performance tests were used. Many of the diodes that were made operated in parametric subharmonic oscillators at a $4\frac{1}{2}$-kmc oscillator frequency. Parametric amplification at 6 kmc was also obtained. These diodes unfortunately have the usual stability and burn-out problems of point contact diodes. Abrupt junction diodes are being made on both p- and n-type base wafers. Technological reasons relating to the metallurgy of the materials make these two types competitive in spite of differences in the mobility of the carriers in the base. In some cases, the doping of the base wafer is varied by diffusion techniques. These diodes have been assembled in the encapsulation previously shown for direct insertion into microstrip line. Typical values of capacitance are 1 to 4 $\mu\mu$f and 1 to 4 ohms series resistance measured at 1 kmc.

Diodes with more sophisticated impurity distributions, involving combinations of diffusion, and alloying in both germanium and gallium arsenide, are under development.

V. Diode Measurement Techniques

The diode measurements program is intended to feed back information about the significant diode parameters to those responsible for diode design, and to provide information for circuit designers and analysts.

The diode characterization technique to be described involves the use of low-frequency measurements to establish some diode parameters and high-frequency measurements to determine others. The constant extension of these techniques to higher frequencies, with such modifications as are necessary, must be contemplated as diode quality improves.

The use of both low-frequency and high-frequency measurements in characterization calls for the establishment of an accurate equivalent circuit such as is shown in Fig. 12. This equivalent circuit contains some elements like C_c, the case capacitance, and L, the lead inductance, whose values are initially established using open- and short-circuited dummy diodes, and which are assumed to retain those values in real diodes. The principal object of diode measurements is to obtain accurate values for the junction capacitance C_0 and the series resistance r_s.

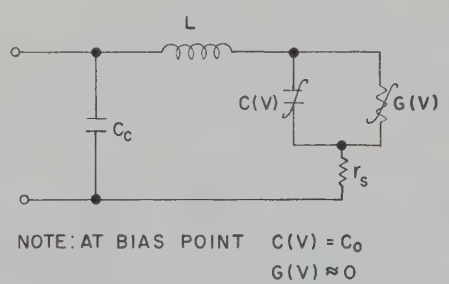


Fig. 12—Equivalent circuit for encapsulated diode.

The basic problem in the measurement of r_s at a frequency below the operating frequency is that a diode which has a good quality factor at the operating frequency will *a fortiori* have an extremely high Q at the measurement frequency. This implies that the capacitive reactance in series with r_s is much greater than r_s;

consequently, r_s is very hard to measure. It is necessary that all other losses (as in the slotted line, diode fixture, etc.) be kept extremely small. While the reactive diode parameters can often be measured at lower frequencies, the series resistance *must* be measured at frequencies near the operating frequency if it is to be measured accurately.

Two measurement techniques are presently being used by the authors. Low-frequency measurements are performed on a Boonton RX Meter. High-frequency measurements are presently made using a coaxial slotted line in the frequency range from 300 mc to 2000 mc.

The RX Meter is an RF bridge with internal provision for an RF supply voltage and null detection. It has a frequency range from 0.5–250 mc and can be read to better than 0.1 $\mu\mu$f. This instrument is used to establish the capacitive parameters in the equivalent circuit at low frequencies, where the relative inductive and resistive impedances are negligible.

A commercially available coaxial slotted line with a micrometer adjustment is used for accurate standing-wave minimum location and measurement. The RF probe voltage is measured using a local oscillator and a 30-mc IF amplifier-detector with about 5 μv sensitivity. This system is useful in the 300-mc to 2-kmc frequency range for determining lead inductance and series resistance. The lead inductance is measured on dummy diode short circuits in which the germanium wafer shown in Fig. 11 is omitted and the wire fastened directly to the metallic post. This parameter is reproducible to within 0.1 mμh from unit to unit. Measurement of the series resistance presents a problem since the over-all Q of the diode is very high at measurement frequencies, resulting in very high standing-wave ratios. The maximum standing-wave ratio accurately measurable on the slotted line is 100–200 in the frequency range from 0.3–2 kmc. This limitation can be restated in terms of the maximum measurable diode quality factor using the chart in Fig. 13. Lines of constant voltage standing-wave ratio and constant quality factor at 1 kmc are plotted against the diode capacitance and series resistance (C_c and L are neglected in this analysis). At any other frequency, it is necessary merely to multiply the capacitance values by the ratio of the frequencies. This chart indicates that, under optimum conditions, it is possible to measure a quality factor of 100 if voltage standing-wave ratios can be measured accurately to 200. For diode impedances which differ from the optimum, smaller quality factors can be measured accurately. On the basis of this chart and preliminary data, it is estimated that 200-kmc diodes represent the upper limit for this slotted line equipment.

Cavity-type measurements have been employed by others to measure diode quality.[5] The design of low-loss cavities (with Q in the thousands) is presently quite feasible. Diode characterization involves determining the degradation of bandwidth and the shift of resonant frequency of the cavity when the diode is introduced. One serious problem in this type of measurement is the need for an accurate resistive dummy diode. The construction of such a dummy resistor is difficult and its measurement presents serious problems.

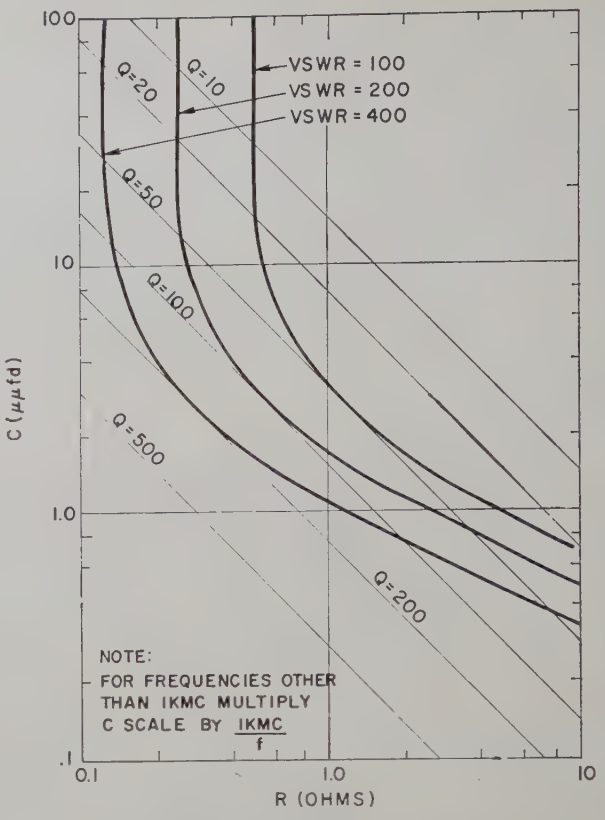


Fig. 13—Standing-wave ratio and quality factor at 1 kmc vs diode parameters.

VI. Discussion

The parametric subharmonic oscillator circuit of Fig. 3 was analyzed by making a simplifying assumption regarding the variation of capacitance with voltage. Only the first two terms in the Taylor series expansion of $C(V)$ were considered. The next two terms have also been examined, and it is found that the conclusions regarding oscillator rise time are pessimistic by perhaps 20 per cent for moderate bias voltages. It is also necessary to point out that the equivalent circuits of Fig. 4 are not really linear circuits. The magnitudes of G_{transfer} and G_{-} are both functions of RF voltage amplitude. However, if the analysis is considered to relate to the energy transfer properties and if the various voltage amplitudes vary slowly, linear circuit analysis may be applied.

The diode encapsulation is of crucial importance in the development of practical diodes. It is desirable that the diode case give rise to as little disturbance of the microstrip transmission line as possible. In the ideal case, where the encapsulation is perfectly matched to the line, the case capacitance will be absorbed into the line (thereby increasing its length) and will not represent an additional independent energy storage to be charged. This ideal can not be achieved. The need for a hermetic seal to protect the diode surface from ambient gases has made large area metallic contacts neces-

sary. This, together with typical high dielectric constants for insulators leads to capacitance values which, indeed, cannot be matched to the common form of microstrip line. In addition, inductance of the metallic finger is too high to permit matching.

The importance of developing a technology for fabricating diodes has been pointed out in this paper. It is, indeed, an overriding consideration in the design of diodes. Series resistance, for example, can be augmented to a very marked degree by the presumed "low resistance" ohmic contacts to each region. The problems in the application of gallium arsenide are primarily those of establishing good junctions of small area and low leakage, and of making good contacts to the bulk regions. Other examples of technological problems are the preparation of small area junctions and of contacts to these areas, the development of suitable etching techniques to prevent the formation of surface channels with high capacitances, the control of diffusion to prevent serious degradation of r_s during the creation of a nonuniform doping distribution, and the development of etching and handling techniques for thinner wafers. These problems will yield to exploration, of course, but they still constitute the greatest barrier in the transition from a practical semiconductor diode to an ideal variable capacitance.

Fast Microwave Logic Circuits*

D. J. BLATTNER† AND F. STERZER†

Summary—**In a carrier-type digital computer system, binary information can be represented by the presence or absence of an RF pulse in a given time interval. Using strip-line printed circuit techniques and point-contact diodes, passive AND and NOT gates were constructed which operate with RF pulses of less than 2 mμsec duration (*i.e.*, an effective pulse repetition rate of 500 mc), at a carrier frequency of 3000 mc. The basic gates were combined to form half-adders. Unlike other carrier approaches, these circuits keep the information in RF form through all steps of the logic operations; *i.e.*, both inputs and outputs of all elements are RF pulses.**

Introduction

In a digital computer operating at an effective pulse repetition rate of several hundred mc, each binary digit is allotted a time interval of only a few millimicroseconds. A baseband system capable of handling such pulses must have a bandwidth extending well into the UHF or even microwave region.[1] An alternative to the baseband system is a carrier system where the required pass band is centered about a suitable carrier frequency.

For operation at a pulse repetition rate of several hundred mc, a carrier system offers significant advantages over baseband systems. It is very difficult to design active and passive components which operate, starting from dc, with the necessary wide bandwidth.[2]

In a carrier system, on the other hand, the design of components is greatly simplified by the fact that, at a high carrier frequency, components with a comparatively small percentage bandwidth have a large absolute bandwidth. Furthermore, carrier-type devices such as traveling-wave amplifiers, parametric amplifiers and oscillators, hybrid rings, ferrite isolators, etc., can be used advantageously in high-speed carrier systems,[3–7] but are not applicable to baseband systems.

Information can be represented in binary form by the presence or absence of dc pulses in "time slots." In such a system, as shown in Fig. 1(a), binary ones are represented by the presence of dc pulses and binary zeros are represented by the absence of such pulses (dc pulse script). Similarly, the presence or absence of RF pulses (RF pulse script) can be used to represent binary infor-

* Manuscript received by the PGEC, March 20, 1959; revised manuscript received June 13, 1959. The work reported here was supported by the Bureau of Ships under Contract No. NObsr-72717. This paper was presented at the 1959 IRE National Convention.

† Electron Tube Div., Radio Corp. of America, Princeton, N. J.

[1] In a baseband system, the signals occupy a frequency band starting at or near zero and extending to an upper limit.

[2] Logic circuits so far reported, using baseband pulses, are limited to pulse repetition rates of about 50 mc (see Walker, *et al.*, and Horton and Anderson). This speed limitation is primarily caused by the limited gain-bandwidth product of the devices used for amplifying the pulses.

R. M. Walker, D. E. Rosenheim, R. A. Lewis, and A. G. Anderson, "An experimental 50-metacycle arithmetic unit," *IBM J. Res. & Dev.*, vol. 1, pp. 257–278; July, 1957.

J. W. Horton and A. G. Anderson, "A full binary adder employing two negative resistance diodes," *IBM J. Res. & Dev.*, vol. 2, pp. 223–231; July, 1958.

[3] W. D. Lewis, "Microwave Logic," presented at the Internatl. Switching Symp., Harvard Univ., Cambridge, Mass.; April, 1957.

[4] W. C. G. Ortel, "Nanosecond Logic by Amplitude Modulation at X-Band," and W. R. Beam, D. J. Blattner, and F. Sterzer, "Microwave Carrier Technique for High-Speed Digital Computing," both presented at the Symp. on Microwave Techniques for Computing Systems, Dept. of Interior, Washington, D. C.; March 12, 1959.

[5] J. von Neumann, "Non-linear capacitance or inductance switching, amplifying, and memory organs," U. S. Patent No. 2,815,488; December, 1957.

[6] E. Goto, "On the application of parametrically excited nonlinear resonators," *J. Elec. Commun. Engrs. (Japan)*, vol. 38, p. 77; October, 1955. Also, "The parametron, a digital computing element which utilizes parametric oscillation," PROC. IRE, vol. 47, pp. 1304–1316; August, 1959.

[7] F. Sterzer, "Microwave parametric subharmonic oscillator for digital computing," PROC. IRE, vol. 47, pp. 1317–1324; August, 1959.

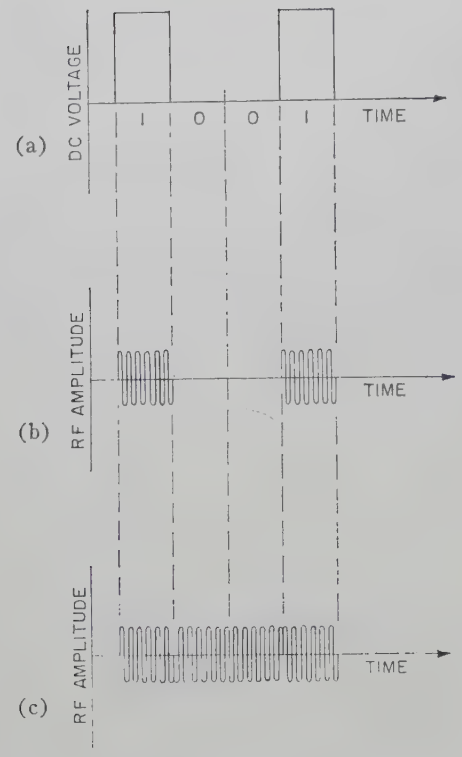


Fig. 1—Three types of script to represent binary information. (a) DC pulse script, (b) RF pulse script, (c) RF phase script.

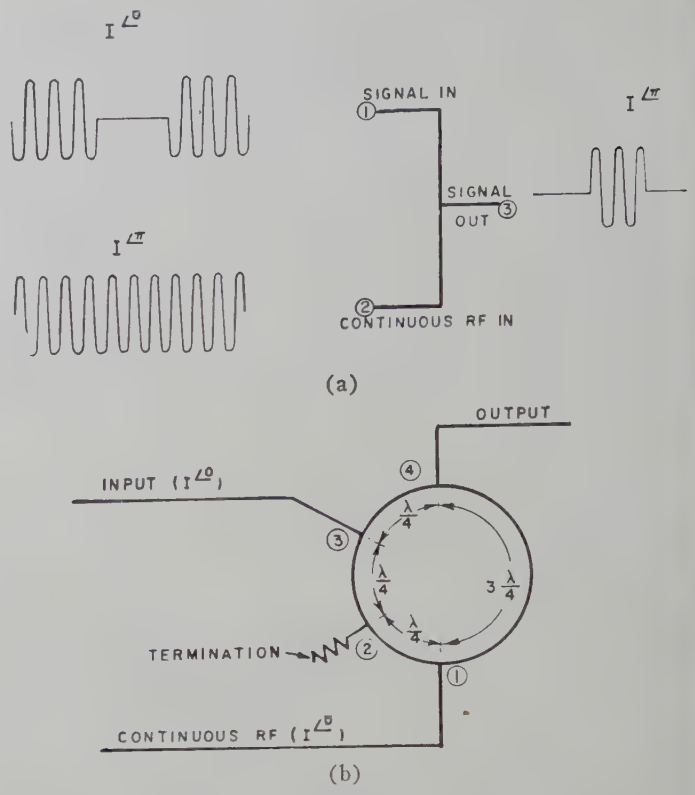


Fig. 2—NOT circuits. (a) A NOT circuit using a simple tee, (b) a NOT circuit using a hybrid ring.

mation, as shown in Fig. 1(b). In another way of coding binary information, shown in Fig. 1(c), a binary one is represented by an RF pulse having a particular phase, and a binary zero is represented by an RF pulse having the same frequency and amplitude but opposite phase (RF phase script). It is possible to convert information from any one of these scripts to any other by means of simple circuits.[4,8] In a complete system it might be convenient to have input and output in dc pulse code, while logic and memory functions are performed in either one or both of the RF codes.

Information coded in phase script can be stored and amplified by use of parametric subharmoninc oscillators.[5–7] Because the several scripts are convertible, subharmonic oscillators can also be used to amplify and store RF information coded in RF pulse script.

This paper describes basic passive logic elements and half-adders using RF pulse coding. All the circuits described were built with strip transmission line;[9] this medium provides compact, low-loss, dispersionless transmission over a wide range of RF frequencies.

Basic Logic Circuits

In principle, any combinatorial logic circuit can be built by using a combination of NOT or complementing circuits and AND circuits.[10] The performance of various designs for these basic logic circuits, using RF pulse coding, is discussed below.

NOT Circuits

Fig. 2(a) shows perhaps the simplest NOT circuit for RF pulse script. The input signal (having amplitude I and phase 0 radians) is fed into arm No. 1 of a simple microwave tee. A continuous signal having the same amplitude, but opposite in phase, is fed into arm No. 2. If there is no input RF pulse fed into arm No. 1 during a given time interval (*i.e.*, a binary zero), there will be an output (representing a binary one) from arm No. 3 caused by the continuous RF input. If, however, there is an input RF pulse applied to arm No. 1, destructive interference between the RF pulse and the continuous RF signal causes complete reflection, and there is no RF output.

One disadvantage of this simple arrangement is that RF power is reflected into the input arm. Although an isolator could be used to absorb the reflected power, it is simpler to use the hybrid-ring circuit shown in Fig. 2(b). The hybrid ring has a circumference equal to one and one-half RF wavelengths and four input arms spaced one-quarter wavelength apart. The input signal is fed into arm No. 3 and continuous RF of the same amplitude, phase, and frequency as the input is fed into arm No. 1. If there is no input, half of the continuous RF is absorbed in the termination of arm No. 2; the other half appears as output at arm No. 4. (Because of destructive interference, no signal leaves the ring at arm No. 3.) If there is an RF input at arm No. 3, the input signal and the continuous RF signal arrive out of

[8] F. Sterzer, "Pulse amplifier with submillimicrosecond rise time," *Rev. Sci. Instr.*, vol. 29, pp. 1133–1135; December, 1958.
[9] M. Arditi, "Characteristics and applications of microstrip for microwave wiring," IRE Trans. on Microwave Theory and Techniques, vol. MTT-3, pp. 31–56; March, 1955.
[10] R. K. Richards, "Arithmetic Operations in Digital Computers," D. Van Nostrand Co., Inc., New York, N. Y.; 1955.

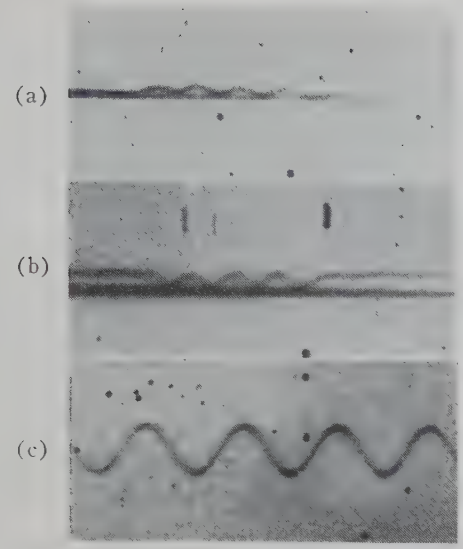


Fig. 3—Oscilloscope tracings illustrating the pulse performance of the NOT circuit. (a) Input pulse train, (b) output pulse train, (c) 200 mc timing signal.

phase at arm No. 4. Consequently, there is no RF output; all of the RF energy is absorbed in the termination in this case.

A stripline version of the circuit shown in Fig. 2(b) was tested with CW RF input, and also with RF pulses having a duration of less than two mμsec. (The RF pulses were produced by a method previously described.[8]) In the CW tests, the RF output signal was reduced by 23 db when an input signal was applied. The results of pulse tests are illustrated by the oscilloscope tracings in Fig. 3. Fig. 3(a) shows the rectified RF input pulses representing the binary digits 1111, together with a base line representing zero RF input. Fig. 3(b) shows the rectified output pulses from the NOT gate; the base line is again visible. Fig. 3(c) shows a 200 mc sine wave used for timing. These three pictures are high-speed photographs of single traces on a traveling-wave oscilloscope.

AND Circuits

An AND gate requires a nonlinear element. Microwave point-contact diodes are suitable nonlinear elements for use with RF pulses.

Fig. 4(a) shows a nonlinear transmission element, called an *expander*, which can be used in an AND circuit. In the expander, a crystal diode, together with a length of stripline, forms an effective quarter-wave stub shunting a transmission line. The diode is biased in the nonconducting direction by means of a battery. If the RF voltage at the diode is insufficient to cause conduction, the diode presents an open circuit to the RF signal. Because this open circuit is a quarter wavelength from the transmission line, an effective short circuit exists across the line, and almost no RF is transmitted. If, however, the RF voltage at the diode is large enough to cause conduction, the RF impedance at the end of the stub is lowered. Therefore, the effective impedance shunting the line is increased, and the RF power appearing at the output is increased. Fig. 4(b) shows the

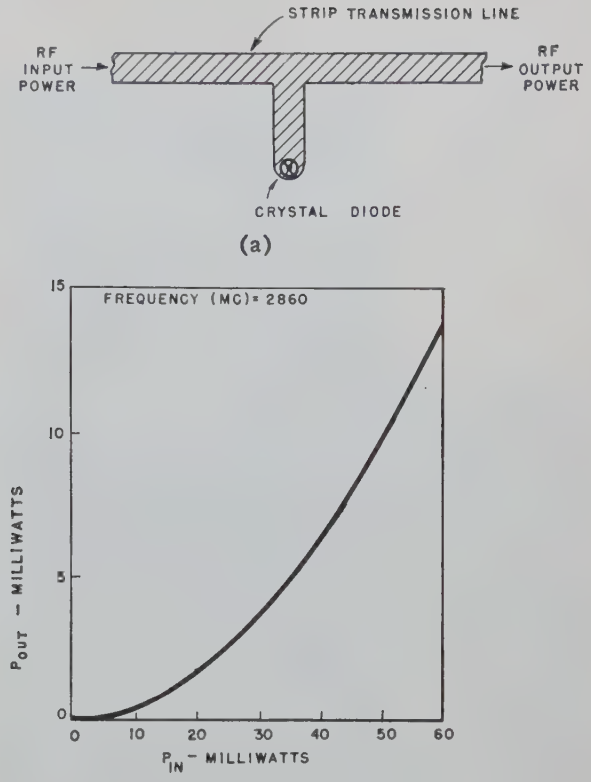


Fig. 4—(a) A stripline expander circuit. (b) RF power output vs RF power input for the expander circuit, using a 1N23D crystal diode with 1.5 volts back bias.

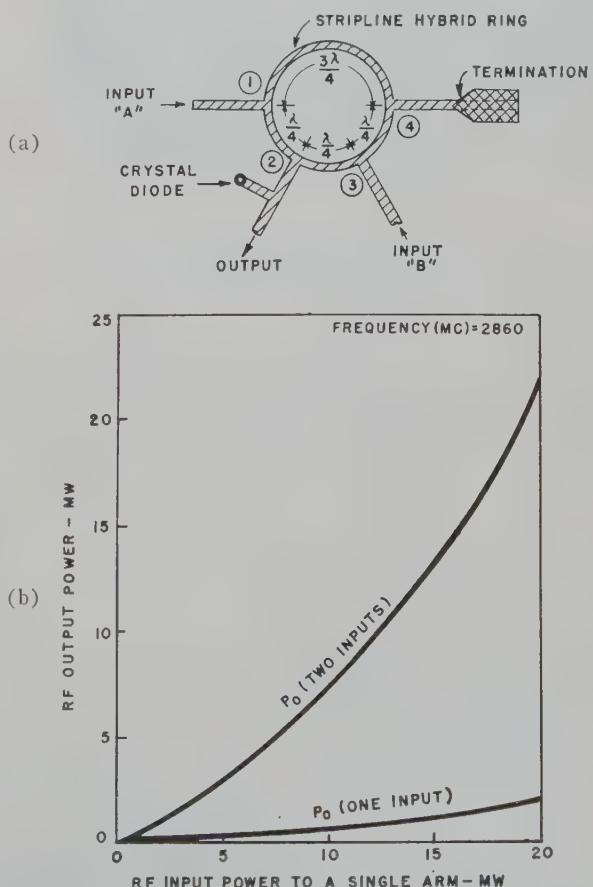


Fig. 5—(a) A stripline AND circuit, using a crystal diode expander circuit; (b) RF performance of AND circuit using a 1N23D crystal diode with 1.5 volts back bias.

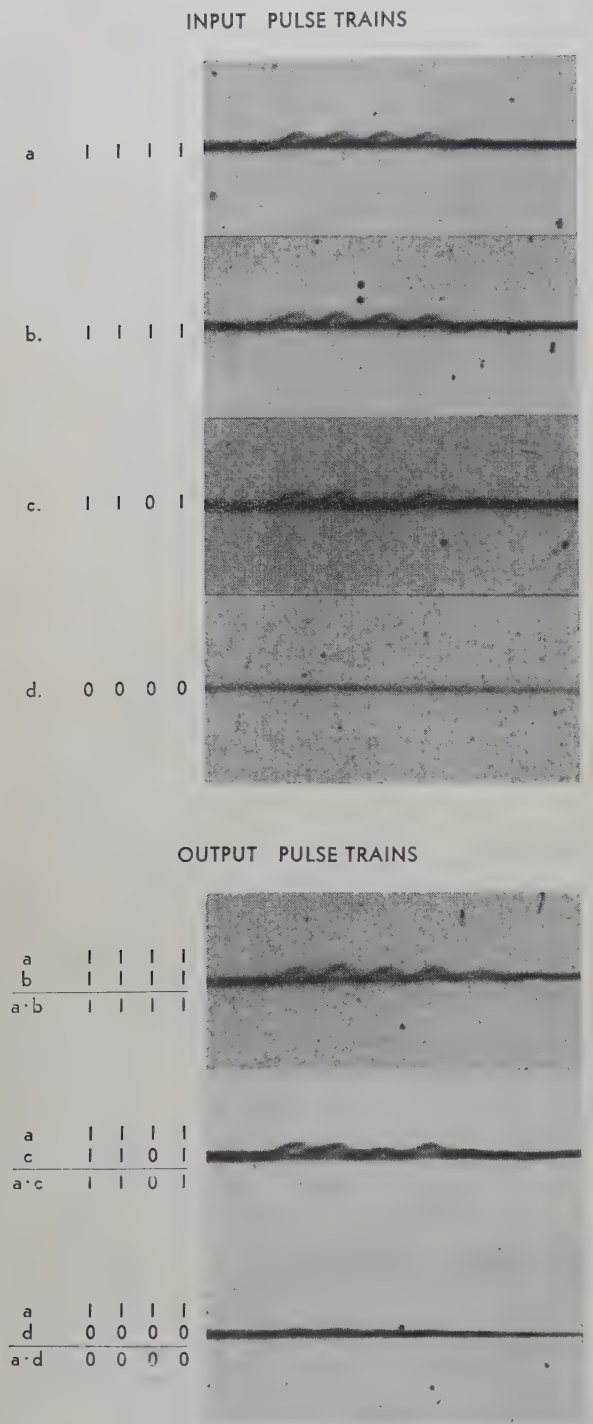


Fig. 6—Oscilloscope tracings illustrating the pulse performance of AND circuit.

performance of the expander at a frequency of 2860 mc. The power transmitted by the expander is plotted as a function of the input power.

Fig. 5(a) shows the construction of the actual AND circuit. An expander is placed in arm No. 2 of a hybrid ring. Arms No. 1 and No. 3 are the input arms, and arm No. 4 is terminated with a resistance card. If only a single input pulse is present, half of its power is absorbed in the termination and the other half appears at the expander. However, if input signals of equal amplitude and phase are applied to both input terminals, they interfere destructively at arm No. 4. In this case no power is lost in the termination, and the signals reinforce one another at the expander. There is, therefore, four times as much RF power incident on the expander as in the case of a single input. Fig. 5(b) shows the CW output power from the AND circuit as a function of the CW input power to a single arm, for one input and for two inputs.

This AND circuit was tested with RF pulses having a duration of slightly less than 2 mμsec. A series of four of these pulses occupied about 8 mμsec. Fig. 6 shows the rectified input pulse trains and the corresponding rectified outputs. Ringing in the circuit is to some extent caused by the modulation, amplification, and demodulation steps required for the oscilloscope display.

Combinations of Basic Circuits: Half-Adders

If the two inputs to a half-adder are denoted as A and B, and the two outputs as SUM (S) and CARRY (C), then the relation between the inputs and outputs is as shown in Table I.[10]

TABLE I
Truth Table for a Half-Adder

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

The AND circuit shown in Fig. 5(a) can be converted into a half-adder if the termination is removed from arm No. 4 and that arm is made the SUM (S) output. Fig. 7 shows the simple half-adder. The output of the expander is the CARRY (C) output. The characteristics of the SUM and CARRY outputs are similar to those of the NOT and AND circuits of Figs. 2(b) and 5, respectively.

A disadvantage of this simple half-adder is that the phase of the SUM output depends on which of the two inputs is present. It is difficult, therefore, to use this circuit in conjunction with following elements which depend upon the phase of RF. One possible method of reestablishing a fixed phase is shown schematically in Fig. 8. Here, an RF pulse of arbitrary phase is demodulated to produce a dc pulse. This dc pulse, in turn, modulates a CW RF signal to produce an RF output pulse. Thus, the phase of the output RF pulse is independent of the phase of the RF input pulse. In conventional microwave modulators, the resistance of microwave point-contact diodes is varied. Such devices have considerable conversion loss, so an amplifier is required. However, it is possible to build modulators with gain by using variable reactances.[11] In this case, no additional amplification is required.

[11] J. M. Manley and H. E. Rowe, "Some general properties of nonlinear elements—part I, general energy relations," Proc. IRE, vol. 44, pp. 904–913; July, 1956.

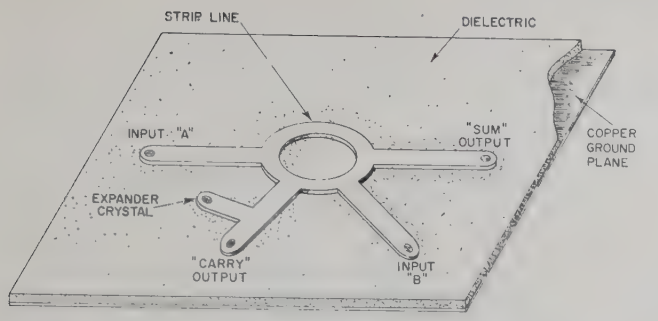


Fig. 7—A half-adder (phase of SUM output not determined).

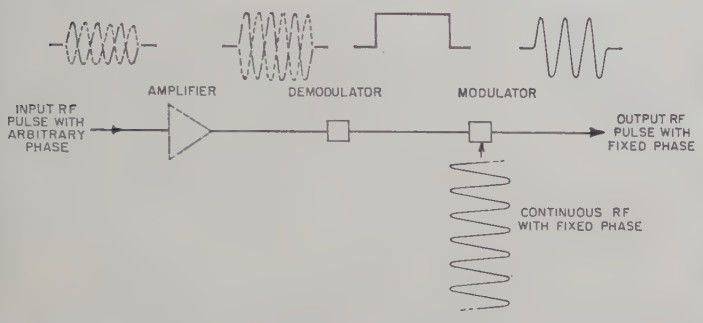


Fig. 8—An amplify-demodulate-modulate circuit for reestablishing a fixed phase of an RF pulse.

It is also possible, however, to build a half-adder in which the phase of the sum output signal is the same regardless of which input signal is present, without resorting to a demodulate-modulate scheme. Such a half-adder is shown schematically in Fig. 9. The RF input signals are divided between two hybrid circuits. The hybrid in the upper left-hand corner simply combines the input pulses. The hybrid in the lower left-hand corner, together with the expander crystal, forms an AND circuit. The output of the AND circuit is split in half. One half appears at the CARRY output terminal. The other half is fed into a third hybrid ring, and arrives at its circumference in phase with the signals from the hybrid in the upper left-hand corner. The adjustable attenuator is set so that these two signals are of equal amplitude when both A and B inputs are present. The third hybrid ring then performs a NOT function to assure that there is no SUM output when inputs are applied to both A and B. The ratios of ONE output amplitude to ZERO output amplitude at the SUM and CARRY terminals are better than 10:1 and 15:1, respectively. Fig. 10 is a photograph of the actual circuit tested; the size of the board is $10\frac{1}{2}\times10\frac{1}{2}$ inches.

Application to Digital Computers

In applications which require more than a few logic operations, the pulses must be amplified and regenerated at intervals. The amplification could be provided by traveling-wave amplifiers, and the regeneration could be accomplished by using a diode regenerator of the type described by DeLange.[12] Another approach is to use

[12] O. E. DeLange, "Experiments on the regeneration of binary microwave pulses," *Bell Sys. Tech. J.*, vol. 35, pp. 1–23; January, 1956.

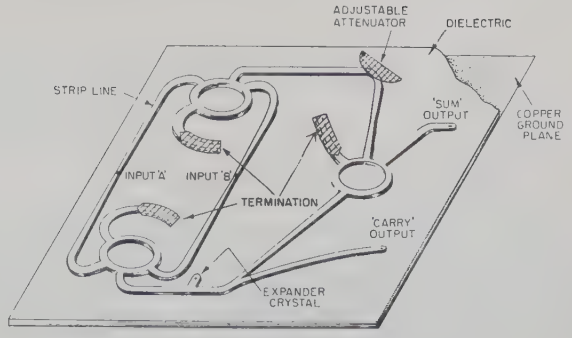


Fig. 9—A phase-determined half-adder.

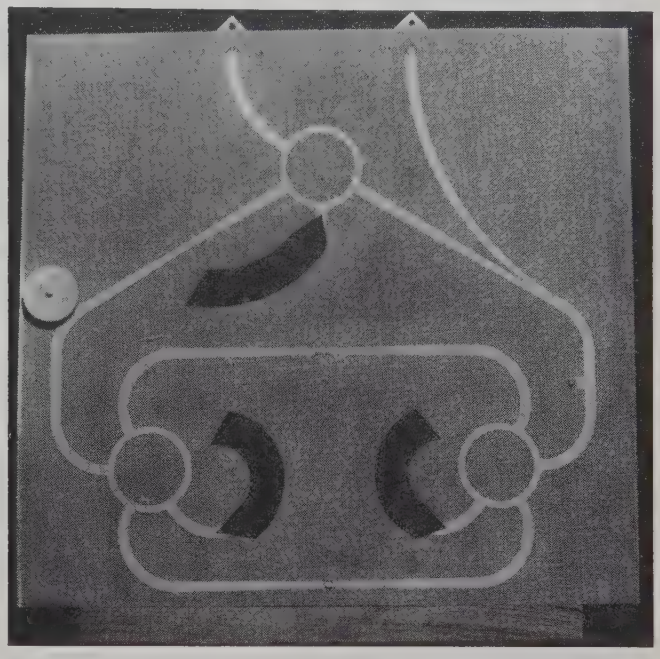


Fig. 10—Photograph of a phase-determined half-adder for use with 3000 mc RF carrier. Coaxial connectors and crystal diode holder mounted on the back of the $10\frac{1}{2}\times10\frac{1}{2}$-inch board are not visible.

subharmonic oscillators, which amplify and regenerate.[5–7] Although subharmonic oscillators themselves can perform logic operations, some of the logic circuits described in this paper are faster and simpler than corresponding circuits using subharmonic oscillators. It is likely, therefore, that a microwave computer could combine both techniques to obtain the advantages offered by each.

Conclusion

This paper has discussed microwave logic circuits which can perform simple logic functions. In operation with a 3000 mc carrier, the pulse repetition rate is presently limited, by the bandwidth of the components, to about 500 million pulses per second; scaling the components to a higher carrier frequency should make it possible to increase the maximum pulse rate substantially.

Acknowledgment

The authors wish to thank Dr. W. R. Beam for helpful discussions and D. L. Thornburg for assistance in the design and construction of the strip line circuits.

Microwave Logic Circuits Using Diodes*

W. SAUTER† AND P. J. ISAACS†

Summary—It is possible to control the transmission of microwave power in a waveguide via external control of the dc bias on a semiconductor diode mounted across the waveguide in a direction parallel to the *E* field. The combination of a microwave detector with such a modulator affords a means whereby RF power in one waveguide can be made to control RF power in a second waveguide. In order to test the applicability of this circuit to binary logic functions, a regenerative memory loop has been constructed. Traveling-wave tubes were employed to raise the level of a controlled signal to that required by the detector. Using an *X*-band carrier, binary pulse stability was observed at pulse repetition rates of 685 mc.

I. Introduction

IT has long been recognized that the effective RF impedance of microwave crystals may be a function of the incident RF power level and furthermore may be a function of the external bias conditions.

Cutler[1] and DeLange[2] have demonstrated that it is possible to make use of the nonlinear impedance of crystals to obtain an amplitude threshold effect in circuits where microwave pulse reshaping is required. While it should be possible to construct microwave logic circuits using two-part nonlinear impedances of this sort, it has been the authors' experience that the nonlinearities that can be obtained are not sufficiently abrupt or not sufficiently broad-band to be useful in millimicrosecond switching.

Numerous references to diode microwave switches[2–6] which perform their function via control of the diode bias have appeared. This paper considers the applicability of such switches to binary logic circuits in which information appears on an AM carrier.

II. Static Switch Characteristics

Investigations at the Diamond Ordnance Fuse Laboratories[5] have demonstrated millimicrosecond switching of *X*-band power using a 1N263 germanium-mixer crystal diode. Fig. 1(a) shows that the insertion loss characteristics of these crystals, when mounted in a conventional crystal mount, exhibit a strong dependence on the external bias conditions and a high degree of uniformity between diode samples. Fig. 1(b) shows the manner in which these characteristics vary with carrier frequency. It can be seen that the ratio of transmitted powers when the diode is forward-biased and zero-biased, respectively, can be expected to exceed 20 db over a frequency range of several hundred megacycles. Thus, if the insertion-loss characteristics are independent of RV power (*i.e.*, the switch is linear), it should be possible to gate broadband signals in the switch.

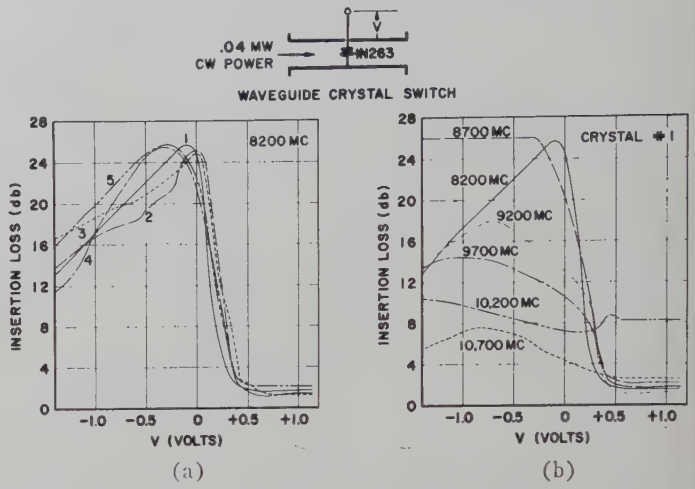


Fig. 1—Crystal switch static measurements. (a) Insertion loss vs bias for random sample of five IN263 crystals. (b) Typical insertion loss vs bias showing frequency dependence.

Fig. 2 shows the insertion loss vs voltage bias at a fixed RF frequency for several RF power levels. These data were obtained using a mount modified to increase the center frequency, but the qualitative results do not differ materially from the results obtained using a conventional crystal mount. The switch may be considered linear for power levels less than several tenths of a milliwatt. The cause of the deterioration of performance at higher powers is related to the rather low value of reverse breakdown voltage of the diode; special germanium diodes manufactured at Philco, which have lower donor impurity concentrations, exhibit linearity over a greater power range.

It may be mentioned in passing that the data of Fig. 2 suggest that the nonlinear impedance of a shorted 1N263 might be useful as a threshold device in RF pulse reshaping circuits. This property has been applied in the construction of an RF pulse memory loop which is similar to a circuit described by Cutler,[1] except that the automatic gain control feature was omitted and, therefore, insertion into the loop of arbitrary pulse patterns was possible. This loop performed satisfactorily

* Manuscript received by the PGEC, April 30, 1959. Presented at the ONR Symp. on Microwave Techniques, Washington, D. C., March 12, 1959.

† Sperry Gyroscope Co., Great Neck, N. Y.

[1] C. C. Cutler, "The regenerative pulse generator," PROC. IRE, vol. 93, pp. 140–148; February, 1955.

[2] O. E. DeLange, "Experiments on the regeneration of binary microwave pulses," *Bell Sys. Tech. J.*, vol. 35, pp. 67–90; January, 1956.

[3] D. J. Grace, "A Microwave Switch Employing Germanium Diodes," Applied Electronics Lab., Stanford University, Stanford, Calif., Tech. Rept. No. 26; January, 1955.

[4] F. S. Coale, "A switch detector circuit," IRE TRANS. ON MICROWAVE THEORY AND TECHNIQUES, vol. MTT-3, pp. 59–61; December, 1955.

[5] R. V. Garver, E. G. Spencer, and R. C. LeCraw, "High speed microwave switching of semiconductors," *J. Appl. Phys.*, vol. 28, p. 1336; November, 1957.

[6] C. A. Burrus, "Millimicrosecond pulses in the millimeter wave region," *Rev. Sci. Instr.*, vol. 28, pp. 1062–1065; December, 1957.

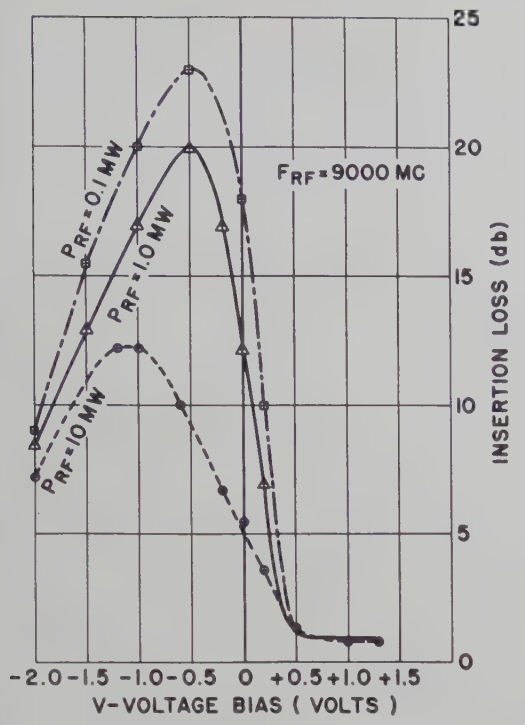


Fig. 2—Effect of RF power level on IN263 diode switch.

at pulse rates of about 250 mc and at pulse pattern repetition rates determined by the delay through the traveling-wave amplifier used.

III. Dynamic Switching Characteristics

Switching-time measurements were made using an unbiased switch. A mercury relay pulse generator having a 0.35-mμsec rise time was used to pulse the 1N263 diode in the forward direction. The 9300-mc RF pulse was detected and displayed on a traveling-wave oscilloscope. Because of the low peak RF power and the limited sensitivity of the scope, the observed deflection provided insufficient information to permit rise-time measurements to be made. A traveling-wave amplifier was therefore used between the switch and detector. The peak-pulse amplitude was limited to the order of 100 mw and, as a result, the detector operated as a linear detector. Rise times (10 per cent and 90 per cent) of 0.7 mμsec were observed. No attempt was made to separate the rise-time contributions of the switch input capacitance, traveling-wave amplifier bandwidth, and detection response from rise-time limitations inherent in the switching crystal. It is believed, however, that since the rectification efficiency of highly doped semiconductor microwave diodes is high up into microwave frequencies, the observed rise time is limited primarily by circuit parasitics rather than by relaxation effects in the germanium. Besides junction heating effects and pulse rise and fall times, pulse repetition rates might possibly be limited by a diode recovery time. That no such limitation exists is confirmed by the fact that when two video pulses are superimposed at the diode terminals in such a way that the time between pulses is controllable by an adjustable delay, there is no discernible effect of the first pulse on the waveform of the second.

High repetition-rate microwave clock pulses may be generated in a diode switch to which sinusoidal modulation is applied. The performance of the switch in such an application has been tested using a 9300-mc carrier and modulation frequencies of up to 100° mc. Oscilloscope observation of the demodulated signal is satisfactory at modulation frequencies up to 250 mc. At higher frequencies, a ridged waveguide microwave interferometer[1] was used. The results indicate that over the range of 250 mc to 1000 mc, at least 80 per cent modulation can be achieved. The accuracy of these measurements suffers, particularly at the higher modulation frequencies, because of the deterioration in the performance of the hybrid junction used to construct the interferometer. These results were checked at a modulation frequency of 960 mc by measuring the signal spectrum components. The first upper and lower sidebands (the only ones measured) were at 5.3 and 5.5 db, respectively, these measurements being accurate to about 0.5 db.

IV. Basic Logic Gate Configurations

Consideration of the characteristics of the diode switch suggests the possibility that the device might find application as a logic element. We note that at low RF power levels, a two-input microwave AND gate can be constructed in the manner of Fig. 3(a). In the experiments to be described in this paper, the demodulator is a crystal detector. To compensate for the conversion loss of the crystal, a traveling-wave amplifier is used ahead of the demodulator since it is assumed that both signals A and B are low level. The maximum power levels of signals A and B are determined by the considerations of Fig. 2.

Fig. 3(b) shows a possible realization of an AND-NOT gate. In this case, the switching diode is forward biased by the voltage supply V_b so that in the absence of a signal A, the insertion loss of the switch is approximately 1.0 db. Other realizations are, of course possible.

In the above basic logic gates, the resistor R serves to decrease the fall time somewhat, and in the case of the AND-NOT gate, to provide a dc return for the bias supply.

The feasibility of logic circuitry based on the microwave diode switch configurations of Fig. 3 cannot be established without considering pulse-amplitude stability and noise susceptibility. Transmission of the "through" signal B will be, at best, linear. Thus, the signal-to-noise ratio of this signal will be, at best, constant. Some means must be found, therefore, for improving the signal-to-noise ratio as this signal passes through successive AND gates. Consider now the "control" signal A. Since a crystal detector is a square law device at low power levels and since the diode switching characteristic is quite abrupt, it might be expected that the signal-to-noise ratio of signal A would be improved.

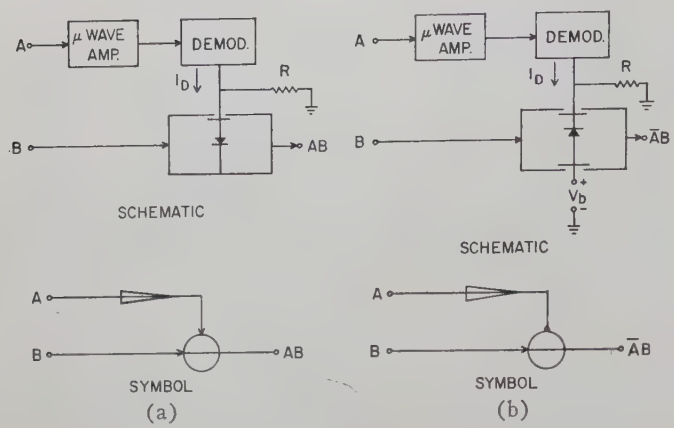
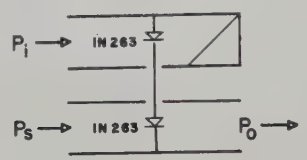


Fig. 3—Basic microwave diode gate configurations. (a) Diode switch AND gate. (b) Diode switch AND-NOT gate.



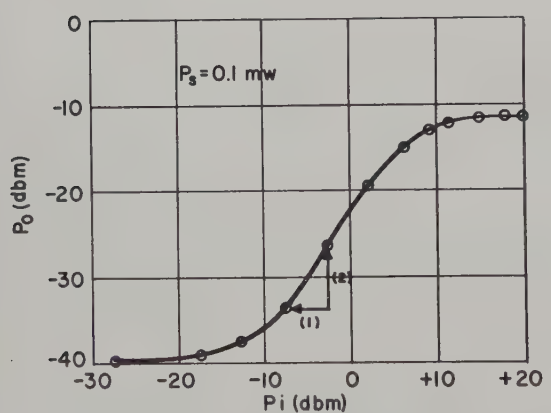


Fig. 4—Power transfer from control input to diode switch output.

CW measurements show that this is indeed the case. Fig. 4 shows a plot of the through power as a function of the control power for a constant power-level input to the switch. A signal-to-noise ratio of 4 db for example, as shown at arrow (1) in Fig. 4, emerges as a signal-to-noise ratio of 6 db at the output of the switch. If a chain of gates is assembled such that the output of each switch is applied through an amplifier to the control input of the succeeding switch, then, with a proper selection of amplifier gain, the signal-to-noise ratio will continue to improve through successive stages until a stable noise amplitude and a stable signal amplitude are reached.

We may now return to the question of stability of the through input. We now have at our disposal a device for pulse regeneration. By applying a degraded signal to the control input of an AND gate, we obtain an improved signal-to-noise ratio. If, in addition, a microwave clock pulse is applied to the through input, pulse retiming and reshaping are simultaneously accomplished. If further improvement in signal-to-noise ratio is required, it is possible to reverse bias the switch diode.

V. Experimental Techniques and Results

In order to test the applicability of the diode switch to microwave logic circuits, a simple microwave memory loop was constructed. Estimates of pulse stability based on static data such as those shown in Fig. 4 afford no indication of pulse stability at a given pulse width or repetition rate. To test pulse stability in the AND gate, a large number of gates could be arranged so that the output of one gate supplies the control signal to a following gate, as shown in Fig. 5(a). Fig. 5(b) shows the required transmission characteristic of the over-all network. A_t represents a threshold above which the test signal stabilizes to the standard level B_1 and below which the test signal attenuates to the stable noise level B_0. The outline of a more practical test circuit is shown in the loop schematic Fig. 5(c), while the details of this circuit are shown in Fig. 6.

The transmission lines which comprise the feedback loop are shown emphasized for clarity in Fig. 6. The amplifier is a special Sperry broad-band traveling-wave amplifier, having a small signal gain of 20 db and a saturation power output of 100 mw. With the amplifier operated in its linear range, the loop gain can be controlled by a single variable attenuator placed anywhere in the loop.

The video circuit linking the crystal detector and the diode switch would customarily consist of coaxial connectors at the output of the detector and at the input to the switch with a length of coaxial cable connecting the two. In the design of this circuit, the following factors must be considered.

1) The crystal detector and diode switch represent nonlinear source and load impedances. While it would be desirable in terms of loop-gain requirements to optimize the transfer of power from the detector crystal to the switch diode, an impedance match here would be only nominal.
2) Lumped capacitance in shunt with the detector or switch and, in view of the mismatched terminations, shunt-distributed capacitance or series-distributed inductance are all detrimental to pulse rise and fall times.
3) A high degree of isolation between the RF circuits of the detector and of the switch must be maintained to prevent signal-to-noise ratio deterioration and to prevent CW oscillations at isolated frequencies within the circuit bandpass for which the loop gain exceeds unity.

In conventional video detector mounts, RF isolation is achieved through the combined effects of an RF choke section and bypass capacitor. This arrangement is frequency sensitive and in most designs excessive shunt capacitance is introduced by the bypass capacitor. The

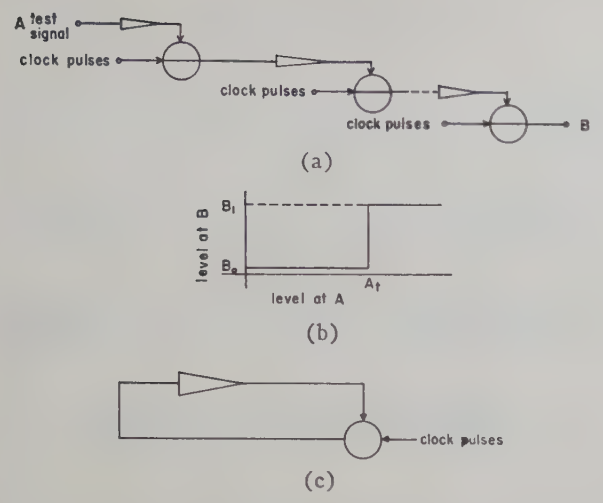


Fig. 5—Diode AND gate test circuits. (a) Iterative AND gate network for testing pulse stability. (b) Desired transmission characteristic of AND gate network. (c) Memory loop for testing pulse stability.

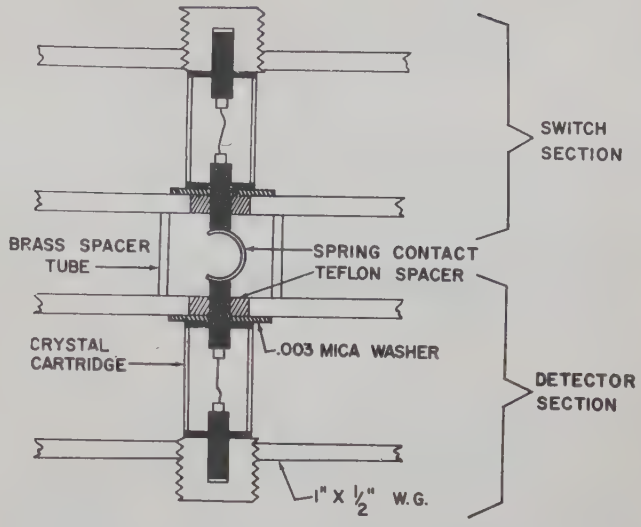


Fig. 7—Low-capacitance detector-switch mount.

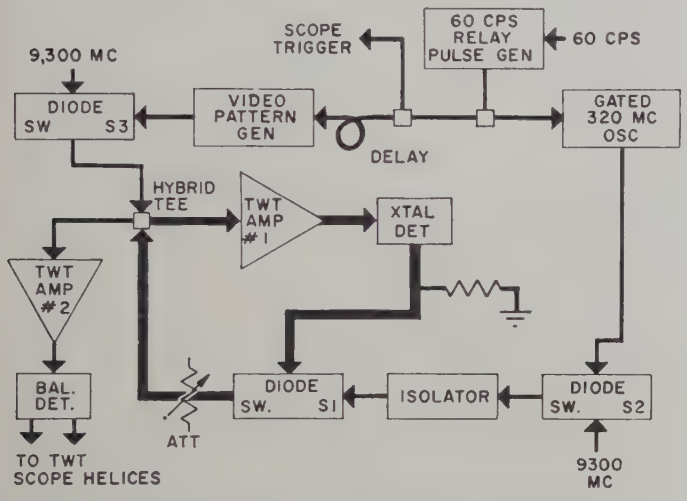


Fig. 6—Memory loop and synchro circuits.

detector-switch design of Fig. 7 relies solely upon capacitance at the waveguide walls to provide isolation between RF circuits. The length of the spacer tube between waveguides can be chosen to provide maximum isolation at the center of the signal spectrum. The design upon which the results of this paper are based is characterized by at least 50 db of isolation over the frequency range from 8200 mc to 10,000 mc, and a shunt capacitance of 2.5 mmfd at the waveguide walls.

In most applications, the detector waveguide section was operated with a resistive termination. The advantage gained through the use of tuning elements in the detector, in terms of reduced conversion loss, was usually not sufficient to compensate for the noticeable video pulse distortion introduced by these elements. Several diode types have been found useful as high-level detectors. Among these, the 1N23WE silicon diode has been preferred for its reversible cartridge style.

Returning to Fig. 6, microwave clock pulses are generated in diode switch S2. A 9300-mc, 0.5-mw CW signal is modulated by the 320-mc output of the gated oscillator. A ferrite isolator is used between S1 and S2 in order that the performance of the circuit be independent of the spacing between S1 and S2.

The input signal for the loop is generated in diode switch S3 and consists of an arbitrary pattern of ones and zeros as determined by the video pattern generator.

In order to assure that the loop input data arrive at S1 in synchronism with the clock, both the video pattern generator and the gated oscillator are triggered via a common pulse source. The repetition rate of this source, a mercury-relay pulse generator, is fixed by the line frequency at 120 cps. At each closure of the mercury relay, a fast-rise rectangular pulse is delivered to the gated oscillator and to the pattern generator and oscilloscope trigger circuits. Since the rise time of the gated oscillator is of the order of 50 mμsec, the loop input data pulses are delayed through a coaxial delay line to allow sufficient time for the oscillator output to reach a steady state.

Proper functioning of the loop depends, in addition, upon coincidence between the data pulses and clock pulses at S1 and thus upon adjustment of the total loop delay to an integral number of clock pulse periods. Upon introduction into the loop of the data pulses, the data will continue to recirculate until the gated oscillator output, and thus the clock pulses, terminate.

The contents of the loop are sampled in the hybrid tee, amplified in the traveling-wave tube (No. 2) and fed to a special double-ended detector which provides a balanced output to the 120-ohm traveling-wave oscilloscope deflection helices.

Fig. 8 shows oscillographs of detected RF pulses in the loop of Fig. 6. The clock pulse repetition rate is 320 mc, and with the total loop length adjusted to the minimum possible number of integral clock pulse periods, the total loop access time was 9.4 mμsec. In Fig. 8(a), a single pulse (100) has been introduced into the loop and is visible as the first pulse in the pulse train. Subsequent traversals of the loop result in a gradual increase in

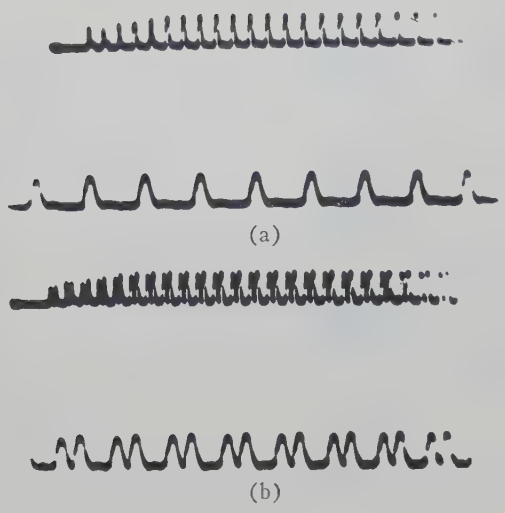


Fig. 8—Loop output: double-pulse (110) input. Oscillographs of detected RF pulses in memory loop showing transient and steady-state conditions. Clock frequency = 320 mc; access time = 9.4 μsec. (a) Loop output: single-pulse (100) input. (b) Loop output: double-pulse (110) input.

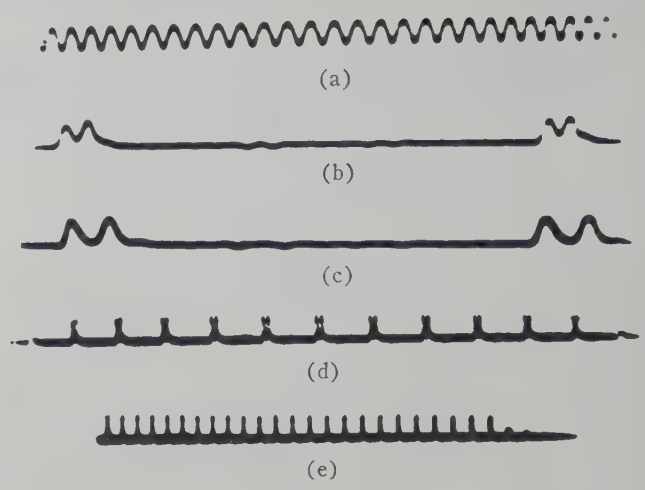


Fig. 9—Oscillographs illustrating operation of 685-mc memory loop.

pulse amplitude, until after about four traversals, a stable pulse amplitude is attained. The rate of convergence to the steady state for signals exceeding the threshold, and for signals less than the threshold, will depend, among other things, upon the characteristics of the input data pulse. Therefore, the transient state as depicted in Fig. 8 should not be construed as being inherent to the loop, but the behavior is typical. Fig. 8(b) shows the loop response to a double pulse input (110).

The lack of dead space between the data pulses of Fig. 8(b) should be noted. It is difficult to determine whether this is due to a limitation on the repetition rate or to limitations in the detection equipment. The oscillographs of Fig. 9, however, which illustrate the operation of a 685-mc loop, suggest that the limitation may be attributed, in part at least, to the detection equipment. Fig. 9(a) shows a 685-mc timing wave. Fig. 9(b) and (c) show the steady-state loop contents to the time scale of Fig. 9(a) when (110000 · ·) and (1010000 · · ·), respectively, have been inserted into the loop. Fig. 9(d) shows the waveform of Fig. 9(c) at a slower sweep speed to illustrate signal stability over many loop traversals. Fig. 9(e) shows the same waveform over the entire period during which the gated oscillator is in operation.

The total loop delay is approximately 32 mμsec as compared to 9.4 mμsec for the 320-mc loop. It was found that 20 db of TWT (traveling-wave tube) gain was not quite sufficient for stable loop operation; approximately 5 db of additional gain was required. The only other amplifier available had a small signal gain of 30 db and introduced an additional 24-mμsec delay.

VI. Applications

The use of this form of diode gating requires the frequent use of amplifiers to raise the level of a through output to that required for a control input. At the present time, the only amplifier capable of providing the proper bandwidth is the traveling-wave tube. Besides being expensive at this time, such a tube introduces a delay which is long compared to the inverse of its bandwidth. This introduces logical complications, since it puts a lower limit on the shortest logical loop. This limit will be several clock times. The lack of a storage unit with a one-clock-time access can be overcome by using a longer loop and storing a number of previous loop inputs in delay lines—the number being equal to the loop delay. The recombination of this information will, of course, require additional gates.

The delay through a traveling-wave tube is proportional to its gain in db. Therefore, the minimum length loop is one in which the TWT just overcomes the losses around the loop. It is desirable, however, that any given signal be capable of controlling several gates as well as providing power for several through inputs. Thus, some compromises between minimum delay and logical gain must be reached. One such compromise would be to have two types of TWT's in the system. It should be possible to design a TWT with a delay of 1 mμsec per 10 db of gain.

The cost of the traveling-wave tube limits the application of this type of gating to those systems in which only a few gates are required to operate at these speeds. One such application is the building of test equipment to provide the signals and detectors to test other types of high-speed circuitry.

A second application would be in a communication system in which it is desirable to make use of the available bandwidth by a high data rate in a single channel, rather than to use many parallel channels. Allied with this type of application is the coding of radar signals either for identification or to reduce the power per pulse.

Another type of application would be used in those systems in which a very simple computation must be done at a high rate. Comparisons or correlations between two binary chains is a typical example in which a small number of gates could be very useful.

VII. Conclusion

The underlying basis for a microwave computer is the belief that higher information rates are possible at microwave frequencies where large absolute bandwidths are available. It would appear that the process of conversion of microwave information to video would severely limit the available signal bandwidth. Nevertheless, it has been demonstrated that logical gating at a rate of at least 700 mc is possible using a video-controlled diode switch; the maximum pulse repetition rate for these circuits has not yet been determined.

The effects of signal delay have been briefly discussed. The requirement for logical gain can be satisfied by increasing loop gain, but only at the expense of signal delay. It is believed, however, that further steps can be taken to increase the logical gain of these circuits without increasing the signal delay.

While the experimental results have not been extended to the construction of any but the simplest of circuits, such an extension is, conceptually at least, very simple. The question of the feasibility of more complicated switching circuits based on the combination of a video detector with a microwave switch must await more quantitative measurements. So far, no attempt has been made to measure the noise susceptibility by introducing external noise, and very little has been done to determine the influence of circuit variations. More practical aspects of the feasibility question include such factors as component life, circuit topology, and cost.

VIII. Acknowledgment

The results described in this paper have been made possible through the support of the U. S. Navy Bureau of Ships. We also acknowledge the contributions of H. Perini, who supplied the data of Fig. 1, and of G. Young, whose practical knowledge of microwave circuits was brought to bear on the solution of many problems that arose during the course of these investigations.

Properties of Propagating Structures with Variable Parameter Elements*

NORMAN KROLL†

Abstract

A MATRIX representation which is particularly useful for the linear analysis of circuits with time-varying parameters will be given and will be applied to the case of a variable capacitance transmission line. The line will first be treated on the assumption that only a signal and a pump frequency ($2\omega_{signal}=\omega_{pump}$) are important, and the conditions which should be satisfied by the respective propagation constants if gain is to be obtained will be presented. If the line is assumed capable of propagating combination frequencies at a velocity close to that of the pump and signal, then the results may be drastically modified. In particular, we will consider the effect of the combination frequency $\omega_{signal}+\omega_{pump}$.

Also treated will be the case where all combination frequencies are allowed to propagate along a dispersionless line.

* Abstract received by PGEC, April 30, 1959; presented at the ONR Symp. on Microwave Techniques, Washington, D. C., March 12, 1959.

† Dept. of Physics, Columbia University, New York, N. Y.; and IBM Res. Center, Yorktown Heights, N. Y.

The Parametron Digital Computer MUSASINO-1*

S. MUROGA† AND K. TAKASHIMA†

Summary—**Features of a large-scale digital computer with novel logical elements, the parametrons, are described. The machine, which is located at Musashino City, Tokyo, was named the MUSASINO-1, and has been in almost continuous operation since its completion in the spring of 1957. Primarily for scientific uses, it does arithmetic operations in parallel, and has a fast access memory of ferrite cores with nonrectangular hysteresis curve. Maintenance experience has indicated its extreme stability and low incidence of faults.**

Introduction

THE "parametron," a new logical element for digital systems on a principle of "a parametrically excited oscillation," was devised by Goto [1], [7], [8] in the Takahashi Laboratory of Tokyo University in 1954.[1] It was considered suitable for digital computers in anticipation of its longevity, and because of high reliability and low cost, compared with fragility of vacuum tubes and costliness of transistors of that time. Our initial objective, set around the beginning of 1955, was to provide at the earliest possible date a large-scale digital computer which would meet the urgent need for various computations in our laboratory. The expensiveness of transistors and fragility of vacuum-tube computers which were available then had been big obstacles to us, but the advent of the parametrons seemed to solve these difficulties, though its operational speed would not satisfy our desires to the full extent.

Our computer was named MUSASINO-1 (usually abbreviated as M-1) after its location in Musashino City, Tokyo. Its continuous operation started when a prototype core memory of small capacity was installed in April, 1957, though freedom in programming was considerably limited. In March, 1958, the memory capacity was enlarged to 256 words, which was somewhat more satisfactory, and since then the computer has been used to its full extent for various programs.

Since its debut as the first large-scale digital computer containing parametrons, it was worked very stably with very few occurrences of faults. It took some time to complete, primarily because of the adoption of entirely new principles in the logical elements and the core memory. Nevertheless, the parametrons have proved to require only easy design techniques. The computer consists of the parametrons of the type made at an earlier stage of development; characteristics of the parametrons have since been improved by Kiyasu and Fukui [2]. The improved model parametron may consume a power of about one-fifth of that consumed by the model incorporated in the M-1 in 1955.

* Manuscript received by the PGEC, February 2, 1959.

† Electrical Communication Lab., Nippon Telegraph and Telephone Public Corp., Tokyo, Japan.

[1] The authors are indebted to a referee's communication that U. S. Patent No. 2,815,488, filed April 28, 1954, issued to J. von Neumann, December 3, 1957, appears to contain material similar to the parametron discussed here.

General Characteristics

The front row of bays of the M-1 is shown in Fig. 1, where sixteen panels of parametrons with a high-frequency-excitation power source, a manual control panel, a magnetic-tape memory and a dc power source are included. The rear row and back of the front row of the bays, with movable cathode-ray-tube monitors hung between them, are shown in Fig. 2. The former consists of the magnetic-core memory and the magnetic-tape memory, and logical wiring over the parametrons is shown in the latter.

The general characteristics of the M-1 are summarized in Table I. The various aspects of the machine may be briefly explained as follows.

Logical Elements

Parametrons are used for the logical elements throughout the computer. About 4400 parametrons are used in the arithmetic and control units, 1000 are used for the control of magnetic-core memory, and 2000 are used for the control of the magnetic-tape memory. The total of about 7500 parametrons is paramerically excited by an alternating current of a frequency of 2.4 mc, which is amplitude-modulated by three-phase rectangular waveforms with a repetition frequency of about 10 kc, superposed on direct current.

Vacuum tubes are used in the high-frequency-excitation current source, in connections between input-output equipment and the arithmetic unit, in connections between the parametrons and neon lamp indicators, and in the distribution of outputs of some parametrons into a large number of successive parametrons. The arithmetic and control units include 280 tubes, and the memory unit of magnetic cores has 239 tubes. The supply of electric current for heaters of these tubes is increased gradually, to minimize deterioration of tubes.

Memories

To meet the requirements on balance of speed with that of the arithmetic unit in a parallel system, magnetic cores were chosen as an internal memory. An ac principle for writing and reading information into or from magnetic cores, which was devised by H. Takahashi and E. Goto in 1955, and which may be considered suitable for a parametron system, was applied on the ferrite cores, combined with a matrix of the parametrons as an address selection circuit. When the magnetic-core

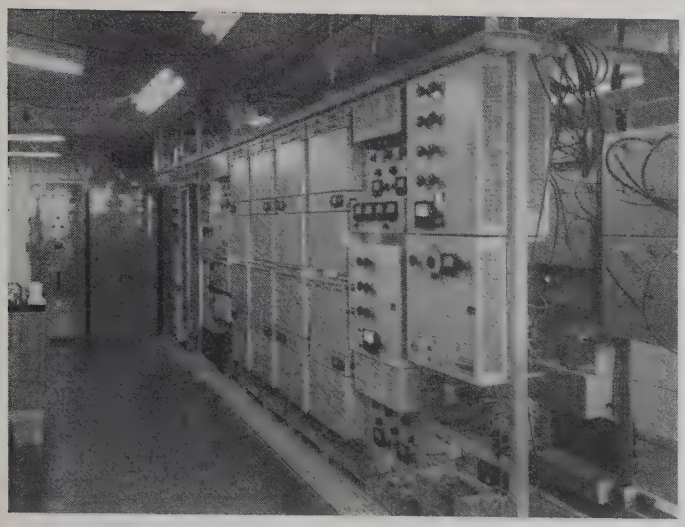


Fig. 1—Front view of M-1.

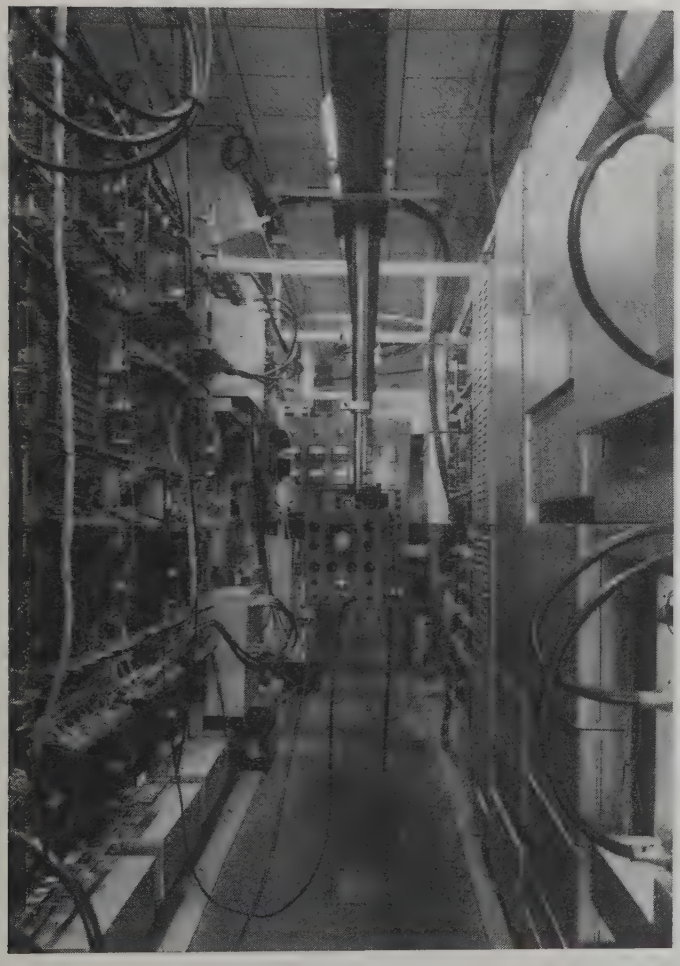


Fig. 2—Midway view of M-1.

TABLE I

SUMMARY OF CHARACTERISTICS OF M-1

Operations; parallel

Representation of number or orders;
a fixed point system with 40 binary digits for a number word and a pair of single address orders for an order word

Number of instructions; about 130

Major components;

Parametrons	
Control unit	1600
Arithmetic unit	2800
Magnetic-core memory	1000
Magnetic-tape memory	2000
Total	7400
Vacuum tubes	
Arithmetic and control units	280
Magnetic-core memory	239
Magnetic-tape memory	160
Total	679

Excitation of parametrons;

Excitation frequency,	2.4 mc
Repetition frequency of three-phase rectangular wave,	6–25 kc
Generation of three-phase rectangular wave,	by flip-flops
Power supply system,	807 push-pull for every 400 parametrons

Magnetic-core memory;

Memory cores,	ferrite toroids with an outer diameter 2 mm and with nonrectangular hysteresis curve
Writing,	by superposition of two currents with different frequencies
Reading,	by generation of the second harmonic of 0.6 mc
Selection circuit,	a matrix of parametrons
Current capacity,	256 words

Required power;

Primary line input,	9 kva
Stabilized output,	5 kw

memory is referred to from the control unit, a row and a column of parametrons in the selection matrix are fed with a current of frequency of 1.2 mc. Consequently, a parametron at the cross point which corresponds to the specified address oscillates because it is parametrically excited with a double amplitude of the high-frequency current, while other parametrons in the row and column do not oscillate because of insufficient amplitude of the excitation. Consequently, a current with a frequency $f/2$ and a constant phase is fed into a wire connected to the selected parametrons, which couples with ferrite cores. When information is to be written in these cores, currents with a frequency f and a binary phase are fed into all the cores, including the cores in which the information is to be written, through 40 wires leading from the arithmetic unit. Each phase of the current is 0 or π radians, corresponding to the digit 0 or 1 of the number word which is to be stored. Then the two currents with the frequencies f and $f/2$ are superposed on the row of the cores corresponding to the selected address, and the information digits are written only on these cores, because a current with a distorted waveform which has a large amplitude on the positive or negative side, corresponding to the digits 0 or 1, results. Conversely, when the stored information is to be read out, a current with the frequency $f/2$ should simply be fed from the selection circuit in the manner described above. Then a voltage with the frequency f, which is the second harmonic of the above current, will be induced in each core in the addressed row, with one of the binary phases corresponding to each 0 or 1 stored

digit. The parametrons which couple with read-out wires will pick up these phases.

The outer diameter of the ferrite magnetic cores is 2 mm. The cores' magnetization characteristic is not rectangular, and so a pair of these cores is used to store a bit of the information. Consequently, requirements on its characteristic are not severe and its cost is relatively low.

The present capacity of the magnetic-core memory, 256 words, is not always sufficient and will be enlarged soon. Magnetic-tape memory compensates for this. The tape memory has been designed to work concurrently with the arithmetic unit and to be free of errors, as far as possible.

Input-Output Equipment

Versatility and reliability for various data-processing tasks, including scientific problems, were considered. Paper tapes with six holes are used for both input and output. One of the six holes is for a parity check, so that the computer stops when an even number of holes is found in reading the tape. Also, a character with an even number of holes types out an unspecified letter when it is put in a printer. The tape reader skips characters with no holes or with all punched holes as special cases.

The printer, with a speed of 9 characters per second, has three shift positions and is capable of printing 78 different characters for tape characters with an odd number of holes. Some of these characters are Arabic numerals, Roman alphabet letters, algebraic symbols, and exponent numerals. A high-speed printer with a speed of about 1500 numeric characters per second has been recently installed. The output can be displayed on a cathode-ray tube in which the brightness of the beam spot can be specified by computer orders.

Representations of Numbers and Orders

To test the versatility of the machine in various research projects in electrical communication fields, besides its use for purely numerical calculations and business accounting, the fixed point binary system seemed convenient. The representation of numbers and orders were patterned, with modifications, after those of the Illiac, because of the designer's familiarity with that machine.

When one word of 40 bits represents a number, the digit which is farthest left represents its sign and the following 39 digits represent its magnitude; the location of the binary point is fixed between the farthest left and second digits. A negative number is expressed as the complement of a two. When one word represents a pair of orders, 20 bits are assigned for each order, with eight bits among them indicating the instruction type of the order, and the remaining 12 digits an address. However, for orders such as transfer of a word block between the magnetic-tape and magnetic-core memories, where a starting address in a core memory and the number of the words to be transferred must be specified, the whole 40 bits are assigned for one order.

Most of the instructions are almost the same as those of the Illiac, but some differences are in the wiring of a 12-bit exchange of address part in memory store orders, the fact that there are no shifts of the accumulator in specified letter punch orders, and the separation of the order for a carriage return and line feed into two orders. New orders incorporated in the M-1 are, for example, four conditional transfer orders on zero in the accumulator, six orders of Boolean operations, five orders for control of the magnetic tape memory, and an order to specify a brightness of the beam spot in the cathode-ray display. Furthermore, if desired, it is quite easy to install new orders simply by changes of wiring among the parametrons.

The Logical Elements, "Parametrons," and the Excitation Source

The M-1's logical element, or "parametron," consists of a pair of toroidal magnetic cores, a capacitor, and a resistor. In other words, it is simply a resonant circuit. The whole set of the parametrons is divided into three groups—I, II, and III. Outputs of the parametrons in group I are coupled to the input transformers of the parametrons in group II, as shown in Fig. 3, but not to those in other groups. Similarly, the outputs of the parametrons in groups II and III are coupled into the inputs of the parametrons in groups III and I, respectively. Into these groups of parametrons, high-frequency sinusoidal currents which are amplitude-modulated by a three-phase rectangular waveform with low repetition frequency, as shown in Fig. 4, are fed, together with direct currents. A parametron in each group is parametrically excited by one of these currents and oscillates to give a sinusoidal voltage with a phase which is synchronized with the input, although the parametron oscillates to give a voltage with either of two anti-phases at random if no input signal exists.

In the M-1's parametron, a ferrite toroidal core, with an outer diameter of 4 mm, an inner diameter of 2 mm, and a thickness of 1 mm, is used, and nine turns of wire are wound over each of a pair of the toroidal cores. The resistor and the capacitor in the parametron circuit are 300 ohms (± 2 per cent) and 5000 picofarads (± 2 per cent), respectively. As a coupling transformer, the ferrite toroidal core with an outer diameter of 14 mm, an inner diameter of 5.5 mm, a thickness of 2 mm, and a winding of 30 turns is used. For excitation of frequency of 2.4 mc, all the parametrons, which were carefully chosen, oscillate to give voltages of frequency of 1.2 mc and of about 3.1 volts within ± 10 per cent allowance [4].

Logical operations are formed by wiring an output of one parametron to another parametron through its transformer by a one-turn winding, as in Fig. 3. The sense of this winding is important; *i.e.*, a winding direction where the phase of the first parametron is conveyed

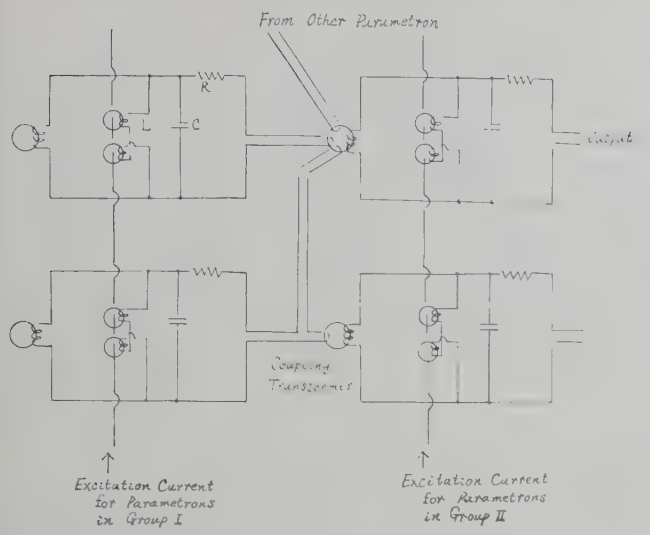


Fig. 3—Parametron circuit.

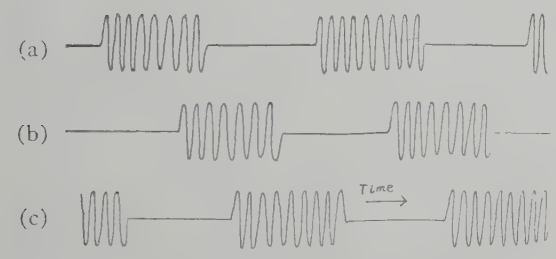


FIG. 4—High-frequency three-phase rectangular waveform. (a), (b), and (c) are currents for the parametrons in groups I, II and III, respectively.

to the second without change is defined as a positive coupling of a variable, and a coupling in the other direction is a negative coupling or negation of the variable. The output of the second parametron is determined by the algebraic sum of these inputs. In other words, the parametrons, which have interesting properties [5], work on a majority principle as do the neurons in the living organisms. For example, if one of three inputs to a parametron is to keep permanently a phase of 0 radians which corresponds to a binary digit 0, an output of the parametron has a phase of 0 radians when either or both of the two variable inputs has a phase of 0 radians. It has a phase of π radians, corresponding to a binary digit 1, only when both of them have phases of π radians. That is, this parametron represents an "AND-circuit." If the constant phase of the parametron is reversed to π radians, it represents an "OR-circuit."

The attenuation of an oscillating voltage (a ratio in decibels of the voltage of the first parametron to that of the second) is 30 db. The larger the number of inputs permitted into one parametron, the fewer parametrons will be required for synthesis of a circuit for a complex logical function [5]. However, the maximum number of the inputs is limited in practice by the nonuniformity of oscillating voltages of the parametrons. In our case, these voltages are adjusted within a range of ± 10 per cent and the maximum number of inputs is therefore limited to five in the arithmetic unit and to three in the control unit.

The maximum number of parametrons in the next stage to which an output of a parametron can be coupled is limited by two factors. The first is variations in phases and amplitudes and of input voltages induced in parametrons in the next stage. If the number of parametrons coupled to one parametron is too large, the whole impedance of serially-connected inputs of these parametrons, which is almost entirely inductive, cannot be ignored when compared with 300 ohms, which is the coupling resistance in the output circuit of the previous parametron stage. Consequently, the phase of the induced voltage will be shifted considerably and its amplitude in each parametron of the next stage will decrease. The induced voltage may not pull each of the parametrons in the next stage into oscillation because of the former effect, and the logic of the parametrons on the majority decision principle may be endangered by both effects because of nonuniformity in the induced voltages. But in practice, the whole input impedance of a number, for instance, 100 parametrons in the next stage, is only 25 ohms for 1.2 mc, which is sufficiently small compared with 300 ohms. This is negligible compared with the factor next discussed, the coupling of input voltages into the parametron from distant ones, jumping over parametrons which intervene between them. Practically, this effect limits the number of branchings from a single parametron. As shown in Fig. 5, where the output of $P_{\mathrm{II}1}$ is conveyed to $P'_{\mathrm{I}i}(i=1, \cdots, N)$ through $P_{\mathrm{III}1}$ and $P_{\mathrm{III}1}$, $P_{\mathrm{III}2}, \cdots$, and $P_{\mathrm{III}M}$, voltages from the $P'_{\mathrm{I}i}$'s are coupled backward into $P_{\mathrm{III}1}$, jumping over $P_{\mathrm{III}1}$, $P_{\mathrm{III}2}$, $\cdots$ and $P_{\mathrm{III}M}$. In addition, a regular voltage from $P_{\mathrm{II}1}$ drives P_{III}. Then the ratio of the voltage from P_{II} to that from the $P'_{\mathrm{I}i}$'s is

$$R = 1/N \cdot 10^{-\alpha/20},$$

where α is a voltage, attenuation per stage in decibels. In the circuit of Fig. 3, we have $\alpha = 30$ db, so if $N = 31.6$, or if the number of the parametrons in the second stage exceeds about 30, there is a danger of malfunction caused by these jump couplings. Considering phase relations beside amplitude ones explained above, safety of logical operations requires that the maximum number of parametrons in the second stage be about 15.

Since M-1 has a parallel-type arithmetic unit with 40 bits per word, there are cases in which it is necessary to control simultaneously 40 or 80 parametrons from a single parametron. In these cases only, buffer amplifiers with vacuum tubes are used.

All the parametrons are driven by the high-frequency currents of 2.4 mc, which are amplitude-modulated by three-phase rectangular waveforms as shown in Fig. 4, with superposition of a direct current after the power amplifiers. Fig. 6 shows a system diagram of this high-frequency current supply. The 2.4-mc oscillator is controlled with a quartz crystal, and modulated with balanced modulators of double 6AS6 tubes. As for the

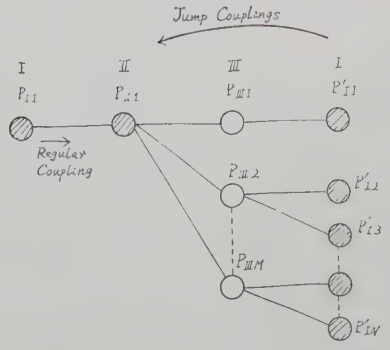


Fig. 5—Jump couplings.

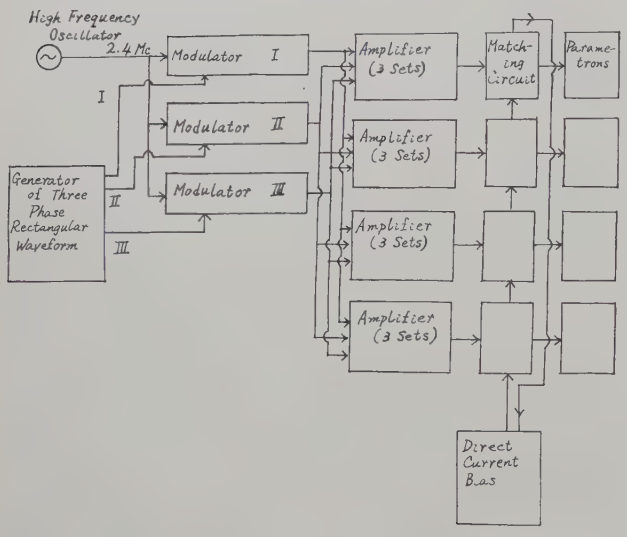
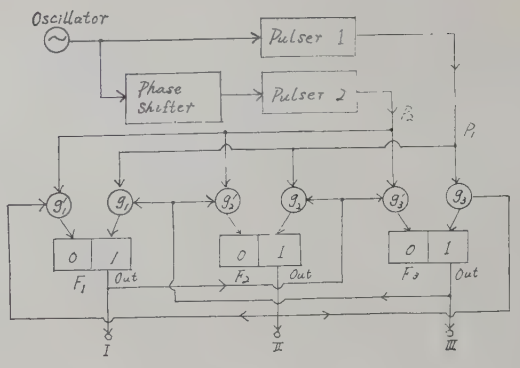


Fig. 6—System of excitation power source.



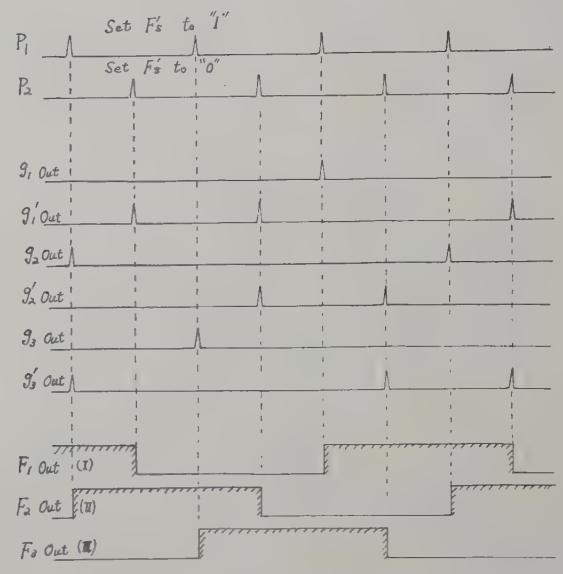


Fig. 7—Generation of three-phase rectangular waveforms.

generation of the three-phase rectangular waveforms, it has often been the practice to limit, by cathode-coupled amplitude limiters, the outputs of three loop-connected amplifiers with a phase shifter interleaved between them, but this method has the disadvantage that variations in the widths of the rectangular waveforms occur with deteriorations of vacuum tubes and changes of power-source voltage. In the M-1, a combination of pulse technique and flip-flops is used instead to obtain better long-term stability. Fig. 7 shows a block diagram of this circuit and waveforms at various points. Two sequences of pulses, P_1 and P_2, are generated at pulsers fed directly and through a phase shifter from an oscillator of a low frequency of around 10 kc. Sequence P_1 is to set other flip-flops to a state "1" and P_2 to a state "0." P_1 and P_2 are coupled to the flip-flops through gates g_1 to g'_3, each of which passes a pulse sequence P_1 to the next flip-flop and P_2 to the previous flip-flop only when the current flip-flop is set in "1." Thus, three-phase rectangular output waveforms of F's are obtained. Overlapping between the rectangular waves can be set at an arbitrary width by changing the interval between P_1 and P_2 by a phase shifter. In this method, the flip-flops are used simply to retain states, so this circuit works very stably as long as the amplitudes of the input pulses to the flip-flops are set large enough. No malfunctions are observed for the ac power source varying from 105 volts to 60 volts.

The power consumption of the parametrons used in the M-1 is 160 mw peak and 80 mw average for three-phase excitation. Thus, a high-frequency power source of 320 watts average or 640 watts peak is required for the 4600 parametrons in the arithmetic and control units. This is roughly 200 watts in peak value per phase if the number of the parametrons in each phase is almost the same. In our case, 807 tubes in parallel supply this power, from reliability considerations and for flexibility for expansion in the future. Each amplifier, in which 6AQ5 tubes in push-pull drive 807's in push-pull in class-B amplification, excites about 400 parametrons; a total of 12 of these amplifiers are used for the arithmetic and control units. The high-frequency power supply for the magnetic-core memory, however, is somewhat different, because the parametrons are not excited in a stationary way but only when the control unit sends a control signal to the memory.

The output of the 807's in push-pull in each amplifier is coupled with two parallel sets of serially-wired 200 parametrons through transformers where each set is tuned by a capacitor in a parallel. Direct current of around 1.0 ampere is superposed in the secondary of

the transformer on the high-frequency current of about 0.8 ampere.

Rise times of envelopes of the rectangular waveforms in Fig. 4, and overlaps between them, are important factors in driving the parametrons. The rise time should be more than about 5 μsec. The ratio of the voltage-sustained period of each rectangular waveform to the full time should be centered at 50 per cent with tolerated range of ±10 per cent, and the overlap about one-third the voltage-sustained period of the rectangular waveforms. Otherwise, stable synchronization among parametrons is difficult.

Logical Circuits

Logical circuits constructed with parametrons have a somewhat different aspect than electronic ones.

Arithmetic Unit

The speed of the arithmetic unit is increased by providing a high-speed multiplier and a carry detector, in addition to parallel operation. Its major part consists of an accumulator, a quotient register, two shift registers, a multiplicand register, a complementer, a multiplier and a Boolean operator. The complementer is located between the memory and the accumulator and has the function of reversing each binary digit of a number transferred from the memory and of adding one to the least significant digit when the number is to be subtracted from the other one in the accumulator. The accumulator, which is an adder itself, and the complementer for one digit are shown in Fig. 8. Hereafter, a binary input with a constant phase 0 or π radians, which corresponds respectively to a binary digit 0 or 1, will be shown as minus and plus inside a circle which represents a parametron. An overflow of the accumulator can be caused by a manual switch to stop the computer when it is necessary. The Boolean operator is located between the memory and the quotient register, and provides a digit-wise logical product, or a logical sum or a modulo 2 sum, of two numbers in the memory and the quotient register, placing it in the quotient register where in the normal case, a quotient after division appears. Two shift registers of double length are attached to the accumulator and to the quotient register to be used for shifting and holding of a number.

The high-speed multiplier for one digit is shown in Fig. 9. A product is serially obtained at the upper point of Fig. 9 for successive transfers of a multiplicand into the multiplier, gated by the least significant digit of a multiplier number which is being shifted. Thus multiplication can be finished within 80 modulation cycles of the parametron excitation.

The carry detector is incorporated to find an end of carries when addition or subtraction is done in the accumulator. It consists simply of an OR-pyramid of 23 parametrons, attached with all carry digits of the accumulator. As it had been shown that an average of the greatest length of successive carry digits in an addition of two binary numbers of n bits is smaller than $\log_2 n$, [6], it is less than five bits for $n=40$. In the accumulator of the M-1, this corresponds to a delay in five parametrons and the carry detector itself contains a five-parametron delay inside, so addition or subtraction requires a total of $3\frac{1}{3}$ modulation cycles. Since $13\frac{1}{3}$ modulation cycles would be required without a carry detector, it yields a reduction of time by a factor of four.

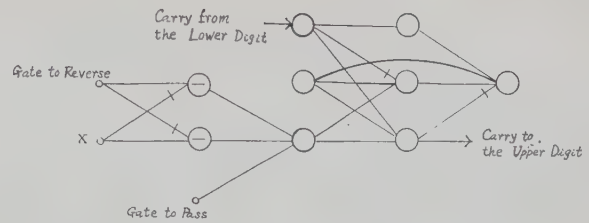


Fig. 8—Accumulator and complementer (one digit).

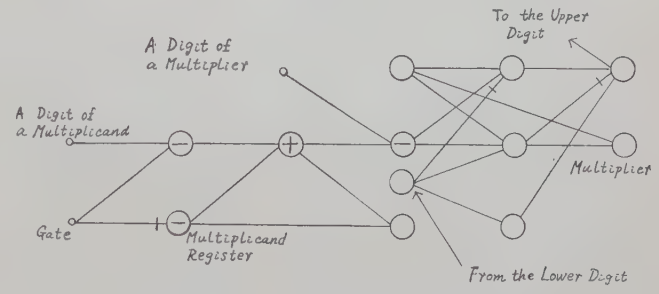


Fig. 9—High-speed multiplier (one digit).

There are about 30 connections from the control unit to the arithmetic unit and a few conversely. Many control signals from the various parts in the control unit gather to each of these connections, from which a control signal must be distributed to many parametrons in the arithmetic unit in parallel. Although the delay may be small for this gathering when the maximum number of inputs to a parametron is granted large enough, three is taken for the maximum for safety, and the consequent delays through the additional required stages are great. This is a factor in slowing down the computer's speed as a whole to some extent. For distribution of the output of a parametron to 40 to 80 other parametrons, a vacuum-tube amplifier is inserted to avoid jump-coupling, and negative feedback is used to minimize effects of variations of the tube characteristics.

Control Unit

Since the control unit is quite complex logically, minimization of labor in debugging was stressed. It is grouped into blocks functionally as far as possible, and the maximum number of inputs to a parametron is limited to three. This emphasis on stability and reliability might have been taken too seriously because the parametrons have proved to have enormous stability. Execution of a sequence of orders is performed in an asynchronous way; in other words, an end of an order initiates the next order.

After execution of every two orders, a pair of orders

are read out from the memory, addressed by the order counter, and the instruction code in one of the two is decoded to actuate one of several control circuits and carry out the instruction.

An addition control circuit is used to control the process of adding or subracting a number in the memory or in the quotient register, to or from the one in the accumulator, taking its absolute value, or its increment, or itself. An end pulse is released from it by the carry detector. A division control circuit is used to control the process of dividing a double-length number of the accumulator and the quotient register by a number in the memory. An increase in speed is provided by the combination of a nonrestoring division and the carry detector. During division, a partial residue is stored in a shift register and, if subtraction is overdone, the residue is transferred to the accumulator. Division by zero or a number smaller than a dividend stops the computer and is indicated by a neon lamp and a speaker alarm. A read-write control circuit is always referred to when the memory is addressed. Access time to the memory is about 500 μsec with 6-kc repetition.

An input-output control circuit is used to control an input tape reader, an output punch, and a cathode-ray-tube display. The input tape reader reads in six-hole characters from the paper tape. If a character has an even number of holes, the computer will be stopped, but if it has an odd number of holes, a sexadecimal character represented by four holes out of six is shifted into the digits of the accumulator which are farthest right. This process is repeated the required number of times using a step counter. Conversely, when the contents of the accumulator are to be punched, a shifted sexadecimal character is punched out with a parity check digit. This process is repeated a required number of times for a normal punching order, where it is not necessary to punch the fifth hole. Orders are also provided to punch required characters, irrelevant of the contents of the accumulator, a number of times specified by their addresses without any effect on the accumulator. This output punch is replaced by the cathode-ray-tube display by use of a switch which provides that the brightness of the beam spot is specified by an order, and two decoded numbers of eight bits are applied to the X and Y axes.

A few other control circuits have rather simple functions and need not be described here.

An end pulse from some of the above control circuits, or from a step counter as it overflows, is released. For nonjump orders, this passes through a regular path to which a manual control switch is coupled in order to interrupt executions of orders or to execute a single order optionally. However, this goes through a transfer path for jump orders, that is, unconditional jumps and conditional jumps on zero or a non-negative number in the accumulator. A manual control switch, to interrupt executions of jump orders optionally, is coupled here.

The computer is stopped by the indication of a neon lamp for various malfunctions of orders; *i.e.*, the input reader's discovery of an even-hole character, a shift of the accumulator by a time of zero, a failure of a parity check in the magnetic-tape memory, or a wrong division. The computer can be restarted by one of the manual control switches, which cancels the neon lamp indication.

The number of modulation cycles required for each of the orders is independent of the repetition frequency. For example, an order of addition from the quotient register requires 16 modulation cycles as the minimum, 34 as the maximum, and 21 as an average for any repetition frequency. As other typical examples, addition, multiplication, division and a transfer require 27, 130, 520 and 9 modulation cycles on an average, respectively. Actual operation times for these are easily calculated by dividing the above modulation cycles by the repetition frequency of the excitation. Since the present repetition frequency is 6 kc, average operation times for addition, multiplication and division, for example, are 4, 22 and 87 msec, respectively. When the repetition frequency is at least tripled in the near future, these operation times will accordingly be one third.

Transfer of Information Between the Parametrons and the Outside Devices, and the dc Power Supply for the Computer

Since the parametrons work on ac, conversion of the high frequency into and from dc is necessary for most of the devices attached to the computer.

Conversion of the high-frequency signal into dc for the paper punch or the photoreader is done by the circuit of Fig. 10. T_1 and T_2 are transformers with three windings. P_c is a parametron with a constant phase "1" and P_v is a "1" for the information. When P_v is "1," a voltage is induced on the output winding of T_1, a negative voltage on G_1 and then V_1 of the flip-flop is "off" and V_2 "on." In this case, no voltage appears on the output of T_2. When P_v is "0," V_1 and V_2 are led to the converse states. Even during the off period of the excitation, V_1 and V_2 hold the same states until a change takes place in a phase of the parametron. For the control of the paper punch, a winding of a relay is inserted in series in the plate circuit of V_2, and for the control of the photoreader, a Schmidt trigger is coupled to V_2.

Conversion of on-off signals of dc into high-frequency signals for the parametrons is done by a magnetic conversion device where variations of the inductances of the ferrite cores are caused by the direct current. In Fig. 11, the output of P_1 is to be coupled to P_2 through three transformers. A voltage induced in the winding T_1 has a phase opposite to the voltage induced in the T_2 and T'_2, both of which are balanced so that the direct circuit cannot affect them on a high-frequency side. When no direct current appears, the sum of the voltages delivered by the T_2 and T'_2 is larger than that delivered by T_1. When direct current grows large, the inductances of the cores of the T_2 and T'_2 diminish and the sum voltage is much smaller than the other. Thus, the

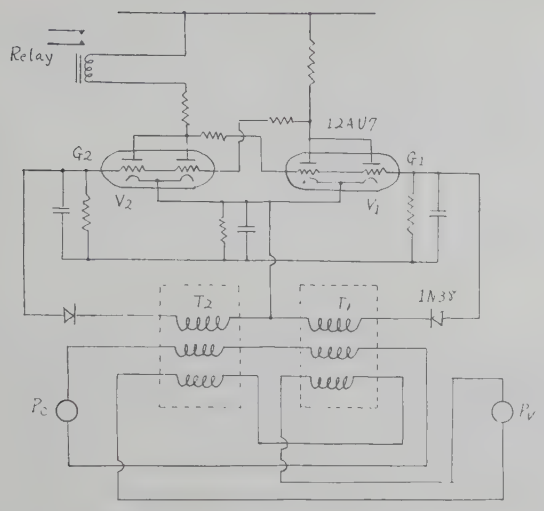


Fig. 10—Conversion of high-frequency signal into dc.

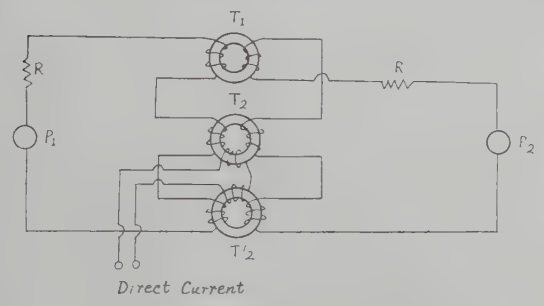


Fig. 11—Conversion of dc signal into high frequency signal.

polarity of the delivered voltage from P_1 to P_2 is reversed by on-off change of the dc level. As ferrite cores are used in this circuit, long-term stability and no deterioration can be anticipated. The cores have the same material and dimensions as those in the parametrons.

The contents of various registers and flip-flops are indicated visually by neon lamps. The serially-connected outputs of the parametron for the constant phase of "1" and of that for the information signal are detected by a diode rectifier and coupled to a vacuum tube which has a neon lamp inserted in its plate circuit. When the latter parametron has a phase of "1," the detected output is zero and then the grid voltage connected to the tube will be sufficient to fire the neon lamp. When the parametron has a phase of "0," a negative voltage is applied to the grid of the tube for a negative rectified output and the plate current is shut off to extinguish the neon lamp. By these neon indicators, the contents of the accumulator, the quotient register, the order pair register, the order counter, stop of the computer, and other situations are visualized for convenience in programming and engineering maintenance.

A power supply of 50-cycle ac is used for the computer, and dc of 100 and 50 volts is used for the paper punch and the printers. For the former, a voltage stabilizer of magnetic amplifier-type is used to stabilize dc at 550 volts, 250 volts, 50 volts, and 30 volts, within ± 5 per cent, ± 2 per cent, and ± 1 per cent, respectively, over a range of 105 to 90 volts in line input voltage of the alternating current.

One-hundred-volt ac for the heaters of the tubes is stabilized within ± 1.5 per cent. The details of the power supply are shown in Table II. The ac input is 9 kva and the stabilized current output consumption is 4.8 kw. This figure of the power consumption could be improved by at least three to five times by use of a new type of the parametrons which have been developed [2].

TABLE II
POWER SUPPLY FOR M-1

100 volts ac (for heaters)	20 amperes	2 kw
550 volts dc	1.5 amperes	0.825 kw
250 volts dc	4.55 amperes	1.14 kw
−50 volts dc	1.8 amperes	0.09 kw
−30 volts dc	7 amperes	0.21 kw
100 volts ac (for others)	5 amperes	0.5 kw
Total		4.76 kw

The whole power supply is turned on and off by two buttons. When it is turned on, the heater voltage of the tubes is automatically increased gradually from a third of the regular value to the full extent in about one minute, and then a bias voltage is supplied. During a run of the computer, if the power supply is interrupted even for a short while, the device alarms with a bell.

OPERATION OF THE M-1

Since beginning of the full operation of the parametrons, about 5500 hours had elapsed to the end of November, 1958, as the computer was operated on the basis of about 10 hours a day from a labor situation. Continuous operation will be started in the near future. No deterioration, except faults by bad soldering, has been found in the parametrons, and high stability and longevity of the parametrons has been confirmed.

During the earlystage of adjustments, various troubles were encountered, and although some of them took some time to fix, there were no essential difficulties. Bad soldering in various places, including the inside of the parametron circuits, was most troublesome. This had caused some difficulties in the adjustment of the logical circuits, because the trouble-free parametrons oscillate in principle, without any input from the troubled parametron which does not oscillate a voltage. But this kind of fault is rather easy to locate because an extinction of the voltage can be found by the cathode-ray-tube monitor. Disconnection of the wires in the coupling transformers may be the hardest fault to locate. In this case, the unsound parametron may oscillate a voltage but the phase signal is not correctly delivered. Therefore, it presents an intermittent fault which may be the most troublesome to pursue. However, once it is fixed it will never deteriorate again. Other types of faults are negligible compared with this. The deterioration of the wires in the coupling transformers seemed to be caused chemically by a soldering paste and somewhat excessive fine-

ness of the wires. As time has elapsed, the number of faults of this kind has largely decreased. There have been none for over half a year since the spring of 1958.

The magnetic-core memory has an excellent stability. No trouble has been experienced with the magnetic cores which are used in a balanced form. The selection circuit which consists of a matrix of the parametrons of a diameter of 13 mm and which has somewhat smaller margins in working stability than the parametrons in the computer, required some adjustments, after which it has presented almost no troubles. The whole magnetic-core memory is divided into four sections. One of the sections was first adjusted carefully and a memory check program was stored there to find addresses with less margins in other sections. This program, which is specifically for the memory, speeded up the discovery and adjustments of the memory, and has been used occasionally for any doubts about the memory.

The vacuum tubes only need to be replaced occasionally. The tubes connected to the neon indicators can be easily found visually, and most of the tubes in the excitation power source which may require maintenance are used with fairly large margins. Replacement of less than 20 tubes is anticipated during a year's time. However, these tubes, except the ones for high-power stages, are being replaced by transistors.

Since deterioration is expected in so few places, no regular maintenance routines on the computer have been applied every day. In ten minutes after supply of the electricity, the computer is available for computations. The input and output equipment is cleaned periodically twice a month, and the tubes are checked every few months.

Sudden changes in the power supply voltages of an alternating current, which cannot be suppressed by the incorporated voltage stabilizer, stop the computer with an alarm, as described above, and those with shorter intervals or minor variations can be detected by a parametron circuit according to the following method. In Fig. 12, three parametrons constitue a flip-flop; coupled to each of them is an unmodulated continuous wave of a frequency of 2.4 mc with an amplitude slightly less than the normal coupling amplitude. If the phase which the flip-flop holds is initially set opposite the one of the modulated wave, the flip-flop will continue to hold the same phase, but a variation in the power supply voltage will drag it into the phase of the unmodulated wave. This will display an alarm with a neon lamp, which signifies to the user the possibility of errors in computation during the run of the computer.

Conclusion

Since completion, the M-1 has contributed to solutions of various problems. The computer was designed in a rather conventional way, but as the parametrons were fairly new elements with unknown factors at that time, the success of the M-1 proved the excellence of the parametrons. There has been no replacement of the parametrons since 1955. Meanwhile, parametrons of improved type with greatly reduced power consumption have promised the possibility of making a computer using only parametrons and transistors, eliminating all vacuum tubes.

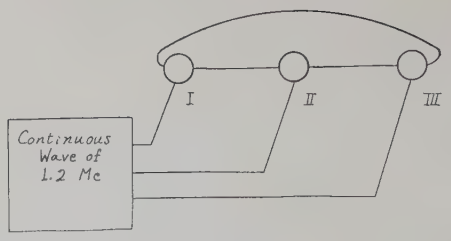


Fig. 12—Parametron circuit to detect voltage variation of a power supply.

Acknowledgment

The M-1 project has been supported by many people. The authors are especially grateful to Z. Kiyasu, chief of Communication Research Section, for his promotion of the project and his supervising interests; to K. Nishida and M. Ono, for their collaboration in the adjustment and maintenance state; to S. Yamada, T. Bessho and T. Koshiba, for the design and completion of the magnetic-core memory; and also to Prof. D. E. Muller of the Digital Computer Laboratory of the University of Illinois, Urbana, Ill., who influenced and guided the authors' ideas. Provision for access to the Illiac by Dr. J. P. Nash, and permission for use of the Illiac Library by the University of Illinois, are acknowledged.

Bibliography

[1] E. Goto, "On the application of parametrically excited nonlinear resonators," *J. Inst. Elec. Commun. Engrs. (Japan)*, vol. 38, pp. 770–775; October, 1955. (In Japanese.)
[2] Z. Kiyasu, "Parametron," *J. Inst. Elec. Commun. Engrs. (Japan)*, vol. 41, pp. 397–403; April, 1958. (In Japanese.)
[3] H. Takahashi and E. Goto, "The memory for parametron computer," presented at meeting of the PGEC, Tokyo, Japan; February, 1956. (In Japanese.)
[4] K. Takashima and K. Nishida, "On the parametron cores," presented at meeting of the PGEC, Tokyo, Japan; October, 1957. (In Japanese.)
[5] S. Muroga, "On the logical elements with majority decision principle and the complexity of their circuits," presented at Natl. Convention of Elec. Commun. Engrs. of Japan, Fukuoka, Japan, May, 1958. (In Japanese.) Though revised papers will be published in English, a part of this report was presented at the Internatl. Congr. on Information Processing, Paris, France; June, 1959.
[6] A. W. Burks, H. H. Goldstine, and J. von Neumann, "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument," Inst. for Advanced Study, Princeton, N. J.; 1946.
[7] S. Muroga, "Elementary principle of the parametron and its application to digital computers," *Datamation*, vol. 4, pp. 31–34; September/October, 1958.
[8] E. Goto, "The parametron, a digital computing element which utilizes parametric oscillation," Proc. IRE, vol. 47, pp. 1304–1316; August, 1959.

A Glow Counting Tube Read-Out Technique and Its Application*

STANLEY K. CHAO†

Summary—**The cold cathode counting tube, also called the decade glow transfer tube, is used principally in preset counter, timing and gating circuits. The number stored in the tube is often required to be recorded. This paper describes a technique whereby the content of the tube is recognized and read out through a carrier signal applied to the anode and 10 detectors connected to the 10 cathodes. The readout is of the nondestructive type since it does not alter the content of the tube.**

A large number of glow tubes can be conveniently read out in this manner simply by connecting all corresponding cathodes together. The carrier signal is then successively distributed to their anodes. An example of such an application is given where 19 channels of four glow tubes each are read into an IBM card punch.

Introduction

THE decade glow counting tube is a cold cathode glow-discharge tube. It has one anode and 30 cathodes.[1–3] Two sets of 10 cathodes are properly connected to become the two guides and the third set of 10 cathodes is used as the output cathodes (Fig. 1). When a high voltage is impressed between the anode and the common cathodes through a current-limiting resistor, a glow discharge is established between the anode and one of the cathodes. The purpose of the guides is twofold: first, they keep the glow on a specific cathode; and second, they guide the glow to transfer in the correct direction. Physically, any number or all of the output cathodes may be brought out of the tube envelope. There are various circuits designed to enforce the transfer of the glow from one cathode to the next.[4,5] Basically, in order to cause the glow transfer, a pair of driving pulses as shown in Fig. 2 must be provided. The guides are usually biased at some voltage more positive than that of the cathode. The driving pulses must be of such amplitudes as to force the guides to go to some voltage more negative than the cathodes. These pulses will force the glow to transfer first to guide 1, then to guide 2, and then on to the next cathode.

The glow counting tube has many uses,[6–10] notably for data storage, as a data accumulator, for preset counting, for timing devices, for gating, and as a select-

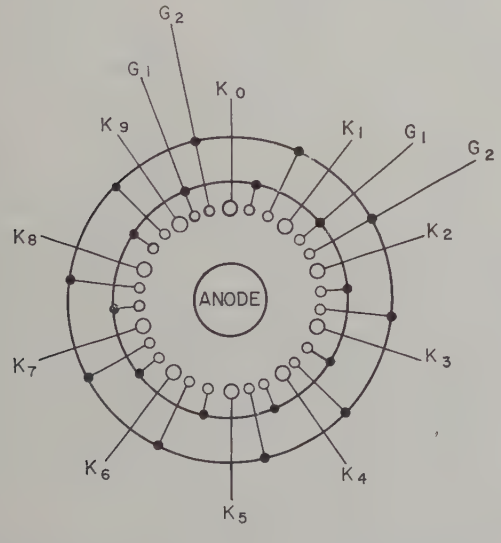


Fig. 1—Cathode arrangement of a decade glow tube.

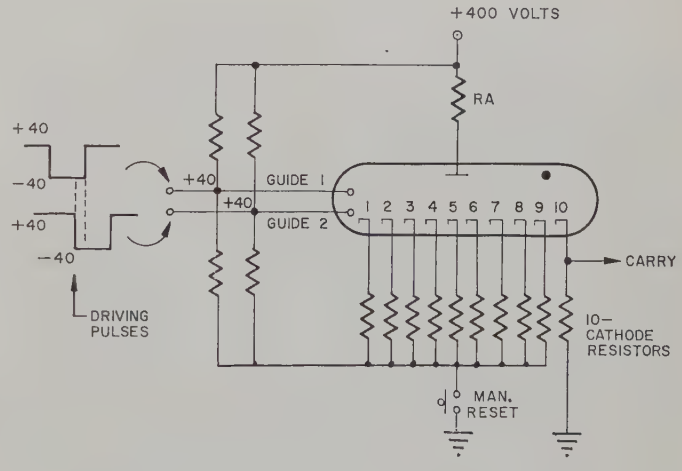


Fig. 2—Schematic diagram of a decade glow tube.

* Manuscript received by the PGEC, March 4, 1959; revised manuscript received, April 17, 1959.

† Data Processing Lab., Sylvania Electric Products, Inc., Needham, Mass. Formerly with Baird-Atomic, Inc., Cambridge, Mass.

[1] R. C. Bacon and J. R. Pollard, "The dekatron, a new cold cathode counting tube," *Electronic Engrg.*, vol. 22, pp. 173–177; May, 1950.

[2] L. C. Brauson, "Reversible dekatron counters," *Electronic Engrg.*, vol. 27, pp. 266–268; June, 1955.

[3] J. Millman and H. Taub, "Pulse and Digital Circuits," McGraw-Hill Book Co., Inc., New York, N. Y., pp. 335–339; 1956.

[4] G. H. Hough and D. S. Ridler, "Some recently developed cold cathode glow discharge tubes and associated circuits," *Electronic Engrg.*, vol. 24, pp. 152–157, April, 1952; vol. 24, pp. 230–235, May, 1953; vol. 25, pp. 272–277, June, 1952.

[5] M. Graham, W. A. Higinbotham, and S. Rankowitz, "Dekatron drive circuit and application," *Rev. Sci. Instr.*, vol. 27, pp. 1059–1061; December, 1956.

[6] R. C. M. Barnes and E. A. Cooke-Yarborough, "An electronic digital computer, using cold cathode counting tube for storage," *Electronic Engrg.*, vol. 23, part I, pp. 286–291, August, 1951; part II, pp. 341–343, September, 1951.

[7] N. B. Acred and G. Bishop, "An electronic timing unit," *Electronic Engrg.*, vol. 28, pp. 276–282; July, 1956.

[8] L. C. Burnett and M. Hawkes, "The dekatron in nuclear instrumentation," *Atomics*, vol. 6, pp. 283–290; September, 1955.

[9] E. L. Harrington, "A high-speed revolution counter," *Electronic Engrg.*, vol. 27, pp. 142–146; April, 1955.

[10] S. K. Chao, "A data processing system using glow tubes," presented at IRE National Convention, New York, N. Y., March, 1959; also see 1959 National Convention Record, pt. 9, pp. 177–182.

ing element. Frequently, the information appearing in the tube needs to be recorded or processed.

Various methods have been used to recognize and process the data stored in these glow tubes. The most common one is to detect the dc level at the cathodes, since a glow on any cathode would cause a current to flow through the resistor on that cathode. A change of dc level of as much as 50 volts is possible. If cathode 4 is 50 volts above the potential of all other cathodes, it is thus recognized that the data in the glow tube comprise a number, 4.

The other frequently-used technique employed to read the number stored in the tube is to send additional pulses into the tube in question. The glow in the tube will advance one position for each pulse sent in. A carry pulse is sent out as the glow goes from cathode 9 to cathode 0. The number of these additional pulses, when counted, can be used to detect the stored number in the glow tube.

The read-out technique described in this paper uses an "anode pulsing" technique. It is so called because the anode of the glow tube is being pulsed to detect the position of the glow-discharge. The usefulness of this technique is demonstrated in a read-out system for a spectrometer, where the data in 19 glow tube accumulators of 4-decimal digits each are detected and read into an IBM summary card punch.

The Anode Pulsing Technique

The anode current in a glow-discharge tube is essentially constant. Since the voltage across the tube after the glow struck will be at a more or less fixed value, the anode current is limited only by the resistor in the anode. In the case of the 4-kc tubes, such as Ericsson's GS10C and Sylvania's 6476, the operating anode current has a range from 250 μa to 550 μa. For a given anode supply voltage, this current can be adjusted by putting different values of resistors in the anode circuit. The anode pulsing technique, used to sense and read the number in the glow tube, utilizes these properties.

As shown in Fig. 3, a "carrier" signal of 5 kc is sent to the anode through the gating tube, V-1. The sinusoidal carrier is always present at the screen grid of V-1, and the control grid is normally gated off. Input pulses going into the glow tube (V-3) would come through the pulse-forming tube, V-2. When the glow tube is selected to be read out, the control grid of V-1 will be gated on. The 5-kc carrier, overdriving V-1, reappears at the anode as clipped sine wave. This waveform is ideal for "pulsing" the anode because of its slow rise time and fall time. The current in the glow tube, when the clipped carrier is applied, varies in the approximate range of 325 μa to 500 μa. This current variation would cause a signal to appear at the cathode where the glow is located. Since a detector circuit is connected to each of the glow tube cathodes, a dc output is thus obtained for the corresponding cathode to indicate the number stored.

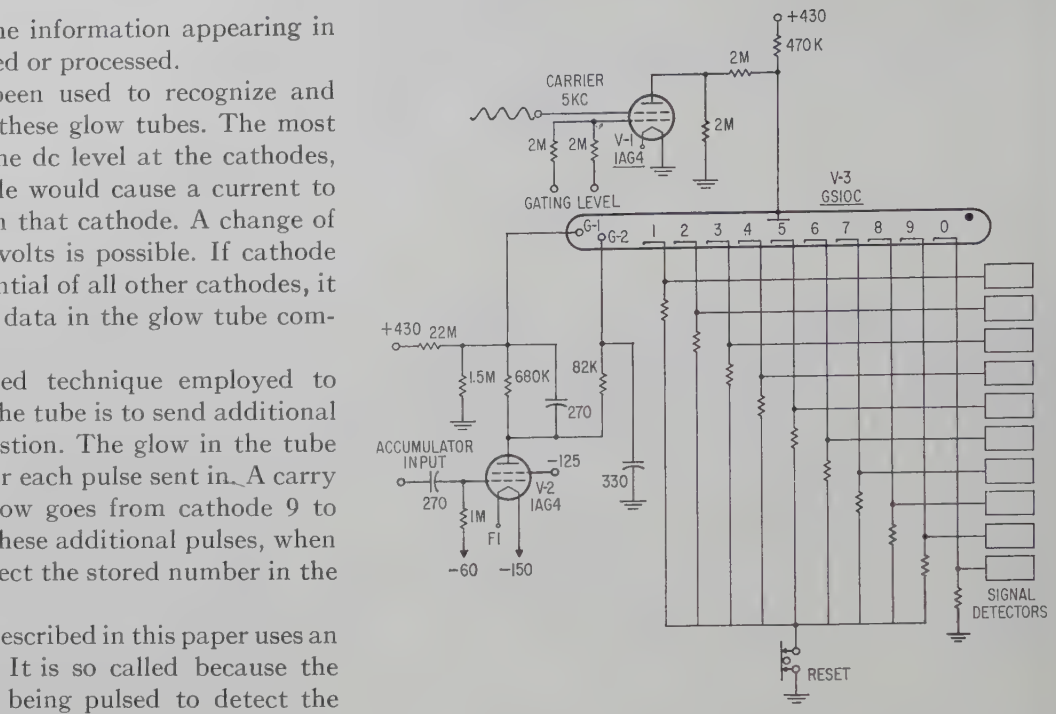


Fig. 3—Anode-pulsing circuit.

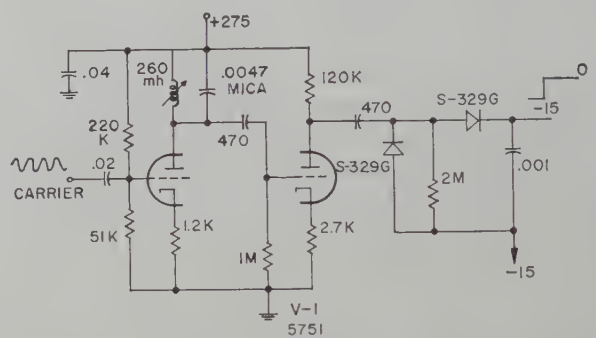


Fig. 4—Detector circuit.

The maximum frequency of the carrier signal is limited by the stray capacity of the glow tube and socket wiring. The stray capacity among the cathodes is determined by the spacings of the cathode-elements inside the tube, the socket wiring, and the total number of glow tubes connected in parallel. If the combined stray capacity is too high for higher carrier frequency, the stray pickup may be so great as to impair the signal-to-noise ratio desired. The lower limit of the frequency for the carrier is the size of the components in the tuned circuit of the detector (see Fig. 4); the lower the carrier frequency, the larger these components.

The principal advantage of this read-out technique is demonstrated when the contents of a large number of glow tubes from various accumulators have to be recorded. As shown in Fig. 5, if there are n glow tubes from various channels and they are to be read out serially, all corresponding cathodes can be connected together, 1 to 1, 2 to 2, etc. There exist only 10 wires for outputs. Each of these ten output wires is returned to ground

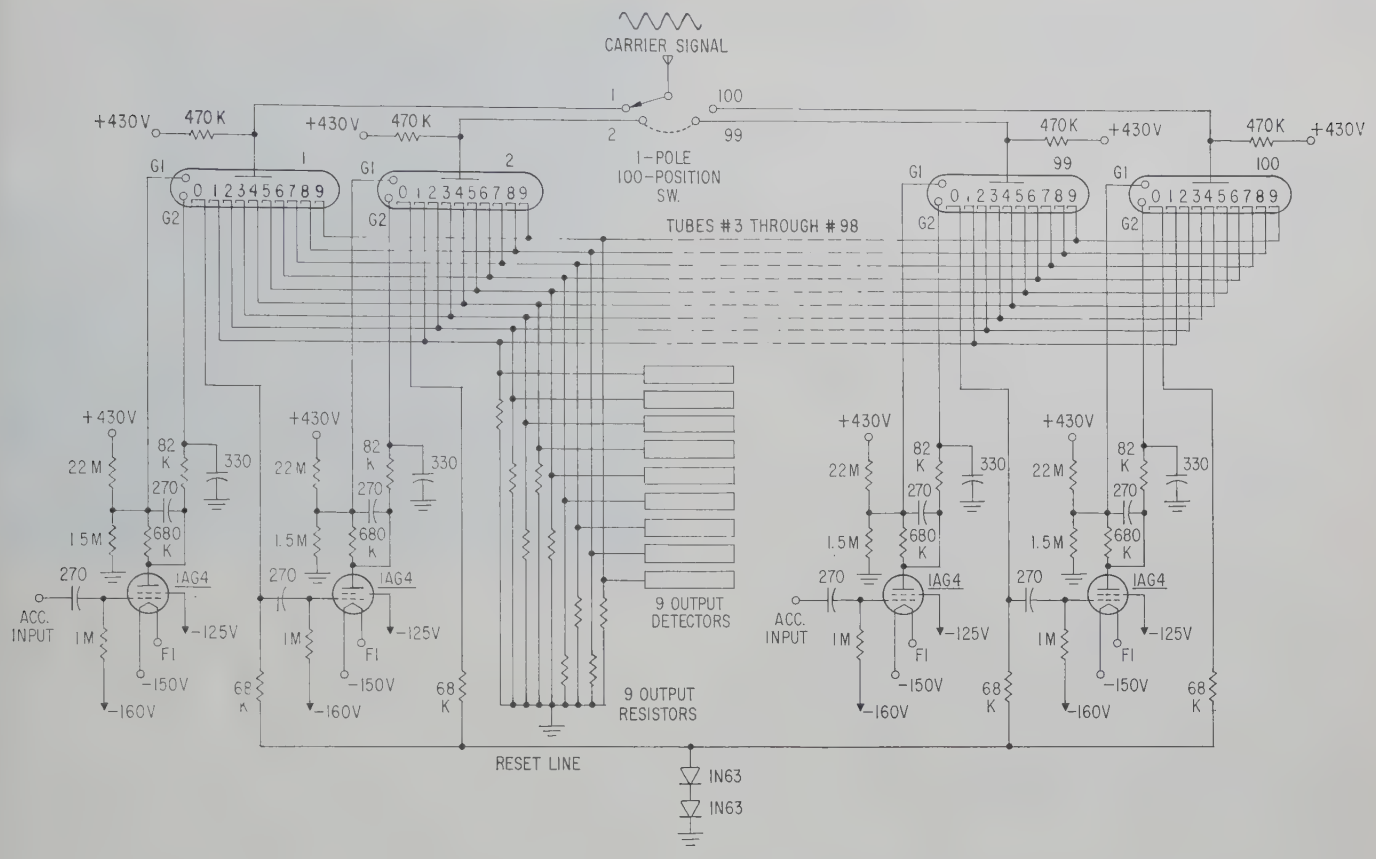


Fig. 5—Anode pulsing circuit for multiple tubes.

through a load resistor. The value of these output resistors is determined by the size of the output signal required. It is also limited by the total number of glow tubes so connected. This latter limitation exists because the quiescent cathode potential should always be below that of the guides.[1] In the extreme case, all n tubes could be at the same number. If each glow contributes a current of 500 μa, 500 n microamperes could flow through the output resistor R_0. The rule is $500\ nR_0 \times 10^{-6} \leq V_{gd}$ where V_{gd} is the guide voltage.

Fig. 5 also shows a carrier distributing switch to direct the signal into various glow tubes. For high-speed switching, gating tubes such as V-1 shown in Fig. 3 could be used. The maximum speed by which the carrier is switched from one glow tube to the next is limited by the output recording device. Of course, the switching speed should be lower than the rate of the carrier signal to allow time for detection in the detector circuit.

Application

As an example of the type of application for which the anode pulsing technique is uniquely suitable, an accumulating read-out system will be described. It was designed for Baird-Atomic's Direct Reading Spectrometer,[11] which is used primarily by the metal manufacturers in the precise analysis of alloys. The light emitted by burning the alloy to be analyzed is decomposed into its constituent wavelengths by the spectrometer. Each line in the resultant spectrum corresponds to an element in the alloy. The line for each wavelength is fed into a photomultiplier which provides an electrical output proportional to the energy received. The ouput of the photomultipler is used to charge a condenser. During the measuring cycle, these condensers are discharged. The measure of percentage concentrations of the elements in the alloy is directly proportional to the discharging time of their respective condensers. Thus, the measurement of element concentration is a measurement of time.

In the read-out system, this time is used to count pulses of standard repetition rate. One accumulator is used for each element to be analyzed. When the measuring cycle starts, the standard 1-kc signal is gated into all accumulators. Individual gates to the accumulators will be turned off as soon as their respective condensers are fully discharged. Consequently, the number accumulated in the accumulators gives an indication of the concentration of the elements contained in the alloy.

Fig. 6 shows a block diagram of the readout system. The system consists of: 1) a 1-kc signal generator, its amplifier and pulse shaper; 2) 18 accumulating channels of four glow tubes each; 3) a 4-digit manual code selector for sample identification; 4) a digit selector to

[11] "Direct Reading Spectrometer," pamphlet by Baird-Atomic, Inc., Cambridge, Mass., No. DR-245; 1958.

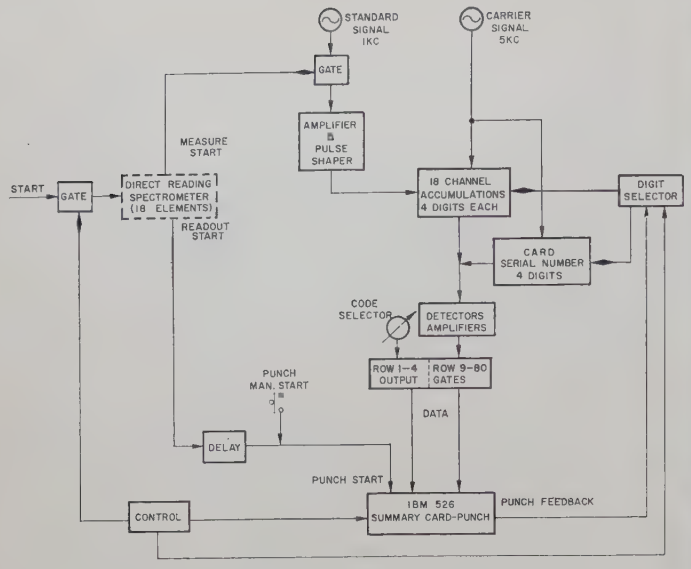


Fig. 6—Block diagram of read-out system for spectrometer.

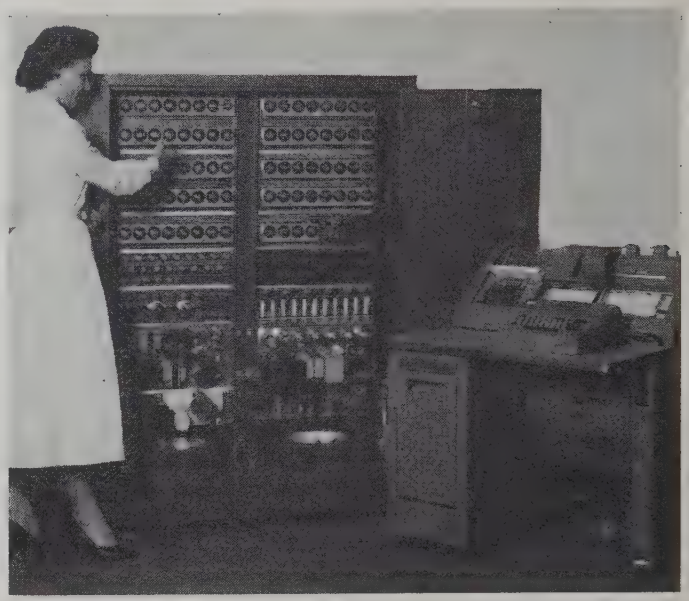


Fig. 7—Read-out system for the National Bureau of Standards.

scan the digits automatically during the read-out operation; 5) a 4-digit glow tube counter to give serial numbers to the cards punched; 6) 10 detectors of the type shown in Fig. 4 for number recognition; 7) an IBM Model 526 card punch to record the data read out of the accumulators; and 8) miscellaneous control and timing circuits.

The accumulators are interrogated by the 5-kc carrier signal, which is serially distributed to the various glow tubes through the control of the digit selector. As explained above, the carrier signal applied to the anode of each glow tube would sense the content of the glow tube and reappear on the cathode having the glow. The detecting and amplifying circuits that follow would cause the corresponding gate to open, thus providing a closed path for the punch pulses to the card punch.

The format of the punched card gives the first four digits as the sample identification, the succeeding 72 digits for the accumulating channels and the last four digits as the serial number. The punching time for the total of 80 digits is approximately five seconds.

Physically, the read-out system (see Fig. 7) was constructed to allow maximum service accessibility. The components are mounted on vertical panels. All panels are fastened on a double rack which is enclosed in a cabinet with plexiglass doors to allow visual observation of the glow tubes.

Conclusion

The readout system has been in operation, with excellent reliability at the National Bureau of Standards since August, 1957. The anode-pulsing technique described is useful regardless of the total number of glow tubes to be read. Of course, the greatest saving in components results when large numbers of glow tubes are involved. Since the number of cathode resistors is always 10, the saving on cathode resistors would be 990 if 100 glow tubes were involved. The digit selecting circuit is also greatly simplified compared with the conventional readout scheme.

Excessive stray capacity will probably impose serious requirements on the associated circuits when the total number of glow tubes connected in parallel is more than 200; however, the series-parallel type of combination can be easily adapted to expand this number, and, if so desired, to increase the read-out speed.

Acknowledgment

The author is grateful to H. F. Stoddart, a Vice-President of Baird-Atomic, Inc., for his invaluable suggestions and his continued guidance throughout the development of this system. Thanks are also due to C. B. Slack who contributed greatly during the initial phase of the development work.

An Idealized Over-All Error-Correcting Digital Computer Having Only an Error-Detecting Combinational Part*

WILLIAM L. KILMER†

Summary—**The block diagram of an idealized over-all error-correcting digital computer is presented. This computer has the property that during each unit time interval, it can correct the effects of a specific maximum number of transient-type component failures which might occur anywhere within it. Yet, all its combinational logic circuitry is only of the error-detecting type. The corresponding reduction in equipment that this design feature makes possible is achieved at the expense of the computer's having to sit idle during a large percentage of those time intervals in which component failures occur. In a sense, therefore, the computer utilizes a great deal of time-domain redundancy as well as equipment-domain redundancy. This paper discusses some of the design requirements that are involved in using this type of redundancy structure.**

Introduction

In the last few years, several separate logical design methods for improving the reliability of digital computers have been presented in the literature. The aim of most of these has been to achieve their end by replacing unreliable operational units by corresponding assemblies of such units, interconnected so as to perform the function of the original units more reliably. The rest of the methods have employed special adaptations of general coding theory.

The purpose of this paper is to describe an idealized computer design which has been the outgrowth of a simple attempt to merge the above two types of methods for improving computer reliability. The operational description of this design emphasizes some principles which have long been known in one form or another by computer engineers, but which have rarely been employed in commercial designs.

This paper will: 1) present the block diagram of the suggested computer design and describe the basic structure of each block; 2) define the computer's operation sequence and state the theorem of the paper; and 3) discuss some of the computer's most important operational and structural properties in a series of informal remarks.

Representation of the Over-All Computer

The idealized computer of this paper is partitioned into conceptual blocks as shown in Fig. 1. There the "combinational logic" block, *CL*, operates continuously with zero delay on the "input" and "memory" output signals. The operation of *CL* is such as to cause it to put out sets of zeros and ones to represent the values of a specified set of Boolean functions. These functions have as their arguments the ordered set of binary variables represented over *CL*'s input lines. Thus, *CL* is said to *realize* a set of Boolean functions from its input signals.

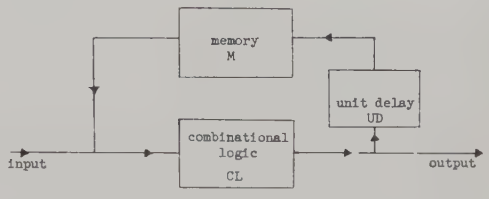


Fig. 1—Block diagram of the over-all computer.

CL's output is used both as the input to the "unit delay" block, *UD*, and as the output of the over-all computer. *UD*'s only function is to pass all its inputs identically after delaying them Δ seconds. Thus, *UD*'s outputs occur continuously throughout all time.

The "memory" block, *M* (see Fig. 1), consists of a set of two-state *memory elements* whose ordered configuration of states at any instant determines the *state* of the memory at that instant. Every variable-value 1 that appears at the output of *UD* is used to either "set" or "reset" a memory element of *M* to the 1 or 0 state, respectively. No variable-value 0 which is put out of *UD* can effect the state of any of the memory elements of *M*. *M*'s state is fed continuously into *CL*, and can be changed instantaneously with the proper changes in *UD*'s 1, 0 output configuration.

It is assumed that the computer is initially put into operation at time $t=0$, and at that time all of *CL*'s and *UD*'s input and output variables are at steady-state 0 and all of *M*'s memory elements are in the 0 state. It is also assumed that the input to the over-all computer is only allowed to change at times $(n\Delta)$ seconds, where n is a positive integer. Thus, the only times any changes can occur in the over-all computer are when t is some integral multiple of Δ seconds.

The Combinational Logic Block

The purpose of this section is to describe the *CL* block of Fig. 1 in greater detail. The order will be to first define the component types from which *CL* is constructed, and then describe the over-all structure of *CL* in terms of an

* Manuscript received by the PGEC, March 9, 1959. This paper is derived in part from a dissertation submitted in partial fulfillment of the requirements for the Ph.D. degree in electrical engineering at the University of Michigan, Ann Arbor, Mich. The actual writing of the paper was supported by Contract No. AF 30(602)-1915, Air Force Res. and Dev. Center, Rome, N. Y.

† Dept. of Electrical Engineering, Montana State College, Bozeman, Mont.

arrangement of subassemblies of these component types.

The component types from which *CL* is constructed are shown symbolically in Fig. 2. All of the x_i inputs to these components are assumed to be binary variables which can take on the value 0 or 1. The output of the component in Fig. 2(a) is the negation of x, denoted x', and is 1 if x is 0 and vice versa. The output in Fig. 2(b) is the conjunct of $x_1, \cdots, x_n$, denoted $x_1 \& x_2 \& \cdots \& x_n$, and is 1 if $x_1, \cdots, x_n$, are all 1, and 0 otherwise. The output in Fig. 2(c) is the disjunct of $x_1, \cdots, x_n$, denoted $x_1 \vee x_2 \vee \cdots \vee x_n$, and is 1 if one or more of $x_1, \cdots, x_n$ are 1, and 0 otherwise. The output in Fig. 2(d) is the majority value among the inputs $x_1, \cdots, x_{2n+1}$, and is denoted $m(x_1, \cdots, x_{2n+1})$.

Component output lines can be split any number of ways for use as inputs to other components, but output lines from two or more components are not allowed to be merged into a single line. There is no operational delay associated with any of the component types in Fig. 2. A component is said to *fail at t* if, during the interval from t to $t+\Delta$, it puts out the negation of the value which it is defined to put out for the input values present during that interval. This type of failure will be the only type considered in this paper. The *cost* of a component is defined as the number of inputs the component has.

At this point, the over-all structure of *CL* can be described. Suppose that the set of Boolean functions realized by *CL* is the set $f_1, \cdots, f_m$, given in conjunction with *N* of Fig. 3. Then, *CL* consists of $k+1$ copies of *N* arranged as shown in Fig. 4. There, each input variable, x_i, to *CL* and each output, f_i, of *CL* is represented over a bundle of $k+1$ lines. Also, the input and output lines of *CL* are connected to the $k+1$ *N* units as indicated by the sub-subscripts on the letter designators of *CL*'s individual input and output lines.

Now consider Fig. 3. It is assumed there that each input variable to *N*, x_i, is represented by a 1 or 0 on the corresponding input line. It is also assumed that *N* is entirely composed of components of the type defined in conjuction with Fig. 2, such that under failure-free operation, the value on each f_i output line of *N* is 1 if the corresponding Boolean function $f_i(x_1, \cdots, x_n)$ is 1, and is 0 otherwise. Finally, the total cost of all the components in *N* is assumed to be a minimum for any assemblage of components having the operational properties just defined for *N*. Hence, *N* is said to be an *irredundant realization* of the functions $f_1, \cdots, f_m$.

It is asumed in Fig. 4 that *in the event that an x_i input bundle or an f_i output bundle has all its $k+1$ lines in either the* 1 *state or the* 0 *state, the bundle is representing either a* 1 *or a* 0, *respectively. Otherwise, the bundle state is not defined.* Obviously then, *CL has the property that if "a" of its components fail at t, and "b" of its input lines are in the wrong state from t to $t+\Delta$, and $(a+b)<k+1$, then none of its f_i output bundles can represent a definitively incorrect output from t to $t+\Delta$.* Thus, *CL* is failure-detecting in this sense, and is called a *k-redundant realization* of the functions $f_1, \cdots, f_m$.

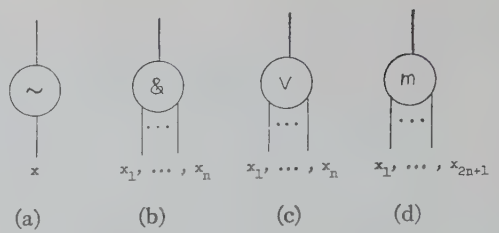


Fig. 2—Component types used in the construction of *CL*.

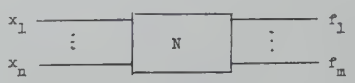


Fig. 3—An irredundant realization of $f_1, \cdots, f_m$.

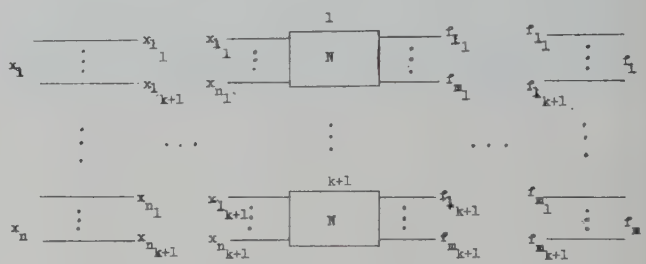


Fig. 4—A *k*-redundant realization of $f_1, \cdots, f_m$.

The Memory Block

M was said to consist of an ordered set of idealized two-state devices called "memory elements." The idea basic to the operation of these elements is set forth in the *primitive memory element* shown in Fig. 5. *There, the "Δ" component is a unit delay element which is never allowed to fail.*[1] Also, the s_i inputs are to set the element's state to 1 whenever a 1 occurs on any s_i line and no 1 occurs on any r_i line. Apparently, if the element's state is set to 1 at t, it will remain there until a 1 occurs at $t+(n\Delta)$, $n>0$, on an r_i input line, provided no failures occur in the interim. Similarly, if the memory element's state is 0, it will remain there until it is set to 1 upon the arrival of a 1 on an s_i line, again barring failures in the interim.

Now consider the schematic for a memory element of *M*, shown in Fig. 6. There, each s_i and r_i is assumed to be an f_i output of *CL* which has passed through *UD*. All the x_{j_i} outputs of the element, shown at the right in Fig. 6, are used to make the jth input to *CL* (the value of j here depending upon which element of *M* Fig. 6 is representing). At this point, assume the following with regard to Fig. 6: 1) there are never any component failures; 2) the same convention is used for interpreting $(k+1)$-line bundle values as was used with *CL* in the section on the combinational logic block; and 3) the majority state of a $(2k+1)$-wire bundle is always regarded as the state of the variable being represented over that bundle. Then, in general, the operation of the ele-

[1] The output response of a set of physical components to a corresponding set of input stimuli must in principle always be delayed whether the response is correct or not.

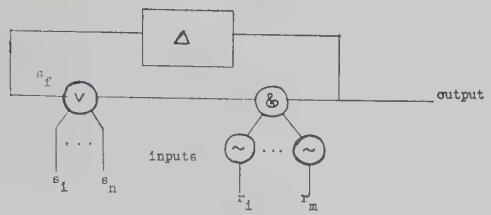


Fig. 5—A primitive memory element.

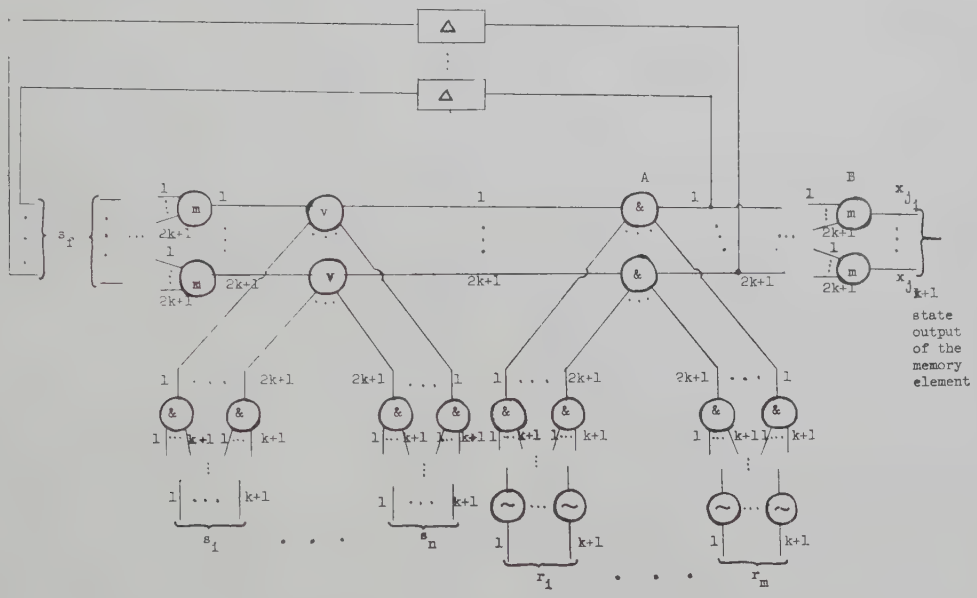


Fig. 6—A memory element of M.

ment in Fig. 6 is the same as that of the element in Fig. 5. This is true since, in Fig. 6, the output state of the element is set to 1 if at some time t a 1 occurs over an s_i bundle and no 1 occurs over any r_i bundle. The output then remains at that value until a 1 arrives over an r_i bundle. This causes the stored 1 which is circulating around the feedback loop to be stopped at the column of "&" components labeled "A" in the figure.

Now suppose that at t, c components in Fig. 6 fail, where $c<k+1$; *i.e.*, suppose that the first assumption in the previous paragraph does not hold now. Then it is easily seen that the value represented over the $(2k+1)$-line bundle (*i.e.*, the majority state among the lines of the bundle) is the same as it would be if no components had failed. It is also apparent that not all of the x_{j_i} output lines can be in the incorrect state. Thus, *if at each time instant, no more than k components of a memory element of M fail, then assuming correct and defined inputs to that element, its state is not affected; also, no more than k of the element's $k+1$ output lines can be in the incorrect state, so a defined incorrect output cannot occur.*

At this point, the paper turns to the detailed operation of the over-all computer.

Detailed Operation of the Over-All Computer

For the purpose of this section, the block diagram of Fig. 1 is further partitioned as shown in Fig. 7. There the M block is subdivided into $M1$, M^*, and $M2$. M^* is a

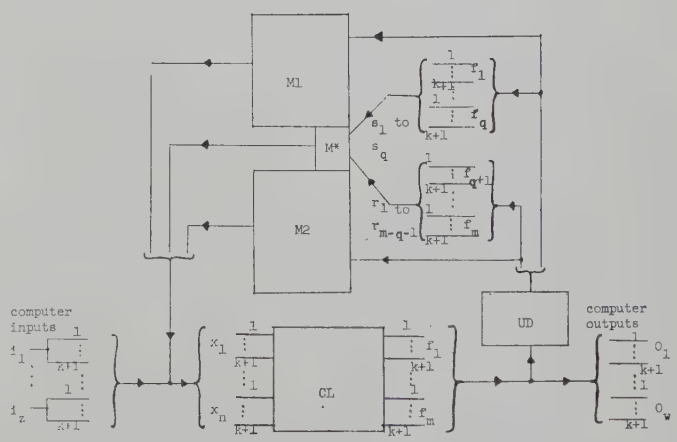


Fig. 7—Refined block diagram of the over-all computer.

single memory element, its s_i inputs being the first q, f_i outputs of CL (delayed by Δ). $M1$ consists of an ordered set of memory elements, each of which has, in general, one s_i and one r_i input coming from among the delayed f_j outputs of CL, $j=1, \cdots, q$. The description of $M2$ is the same as that of $M1$ except that for $M2$ $j=q+1, \cdots, m$.

The $O_1, \cdots, O_w$ outputs of the over-all computer consist of a subset of the $f_1, \cdots, f_m$ outputs of CL. The $i_1, \cdots, i_z$ inputs to the over-all computer, together with the outputs of $M1$, M^*, $M2$, constitute the $x_1, \cdots, x_n$ inputs to CL.

After the computer in Fig. 7 is started, its operating sequence, starting at t and *assuming no component failures*, is as follows: *one* of CL's output bundles, f_j, for example, carries all 1's and the rest carry all 0's. If $j<q+1$, at $t+\Delta$, M^* is set to 1 and the state of one of $M1$'s memory elements is respecified (*i.e.*, set or reset, regardless of its previous state). The setting of M^* to 1 causes CL to transmit a 1 over one of its $f_{q+1}, \cdots, f_m$ output bundles. Then at $t+(2\Delta)$, this output resets M^* to 0 and respecifies the state of one of $M2$'s memory elements. The operation cycle is completed when CL, sensing M^*'s 0 state, transmits a 1 over one of its $f_1, \cdots, f_q$ output bundles.

If, on the other hand, a total of "c" components, $0<c<k+1$, fails throughout the over-all computer at t, there are two cases to consider. 1) If all c failures occur in M, not including any of M's output m components (see the B column of components in Fig. 6), at $t+\Delta$, the operation of the computer proceeds exactly as if c had been 0. The reason for this is stated at the end of the section on the memory block. 2) Otherwise, at $t+\Delta$: a) M^*'s state is not changed; b) none of $M1$'s or $M2$'s memory elements have their states respecified; and c) CL attempts to produce the same output it would have produced at t if no components had failed then. The reasons for a) and b) follow immediately from the fact that a combination of a 1's over an s_i or r_i bundle, and b input & component failures in the corresponding memory elements, where $(a+b)$ is at least $k+1$, is necessary for the state of a memory element to be respecified. Finally, c) is simply a consequence of a) and b). Thus in case 2), the component failures effectively cause the computer to lose one interval of calculating time. But this loss enables the computer to prevent itself from making a permanent mistake.

At this point, note that the computer can have a starting mode as follows. The first input set, $i_1, \cdots\cdots, i_z$, is stored bit-by-bit in $M1$ and $M2$. Then an O_j output 1 is produced to signal that the first block of inputs has been stored (an output is recognized when all $k+1$ lines of the corresponding output bundle are 1). Next, another input set is stored—and so on, until at some point, the computer does not call for any more inputs until it has finished processing some of the information it has already stored.

Some Structural Properties of the Over-All Computer

Let the computer that has been described in connection with Fig. 7 be called C. Then the results of the foregoing are summarized with the following.

Theorem:

If during any time interval from t to $t+\Delta$ the total number of component failures occurring anywhere in C is q, where $0<q<k+1$, then either: 1) *C's operation proceeds at $t+\Delta$ exactly as it would have if q had been* 0; *or* 2) *the memory state of C remains unaltered from $t+\Delta$ to $t+2\Delta$, and the combinational part of C attempts to produce the same output from $t+\Delta$ to $t+2\Delta$ as it would have produced from t to $t+\Delta$ if q had been* 0. *Furthermore, if only the O_j output bundles of C which have all their $k+1$ separate lines carrying* 1's *are regarded as producing "valid" outputs of C, then C can never produce an incorrect sequence of valid outputs unless more than k components fail during some time interval.*

At this point, there follow some remarks concerning the structural properties of C.

Remark No. 1

The memory elements of M are constructed so as to *correct* the effects of any set of k or fewer component failures which occur in C at any time t. This feature is absolutely necessary if the effects of such failures are not to accumulate and eventually reduce the memory state of C to something approximating a random configuration.

Remark No. 2

CL is constructed so as to calculate only one bit at a time to transmit to M. If more than one bit is permitted, as well as up to k component failures per time interval for a particular k, at first there is a possibility that only *some* of the bits calculated by CL each time interval might be properly registered in M. In order to eliminate this possibility, C must be designed to make sure that the state of M at each point is at least closer (in some distance sense—perhaps the Hamming sense[2]) to the desired "perfect information state of M" than any other "perfect information state of M," where the "perfect information states of M" are those states of M which result from failure-free computations.

Remark No. 3

M is partitioned into $M1$, M^*, and $M2$. It can easily be shown that if CL computes bit-by-bit into a *single* memory block, the number of 1's stored in this block must either increase or decrease monotonically with time. And, of course, one could not permit this.

On the other hand, if M is partitioned into two blocks, there must also be a third separate memory element which, by its state, always indicates which of the two blocks the combinational part of the computer is to work into; thus, the memory partitioning for C.

Finally, it can be shown that partitioning the memory into u sub-blocks, where $u>2$, does not lead to anything essentially different from the case where $u=2$.

An Example of C Operating as a Serial Digital Computer

Remark No. 4

In principle, C can be made to operate in essentially the same way as a serial-type digital computer. To illustrate this, let C be required to execute in order the sequence of coded instructions stored in $M1$'s memory

[2] R. W. Hamming, "Error detecting and error correcting codes," *Bell Sys. Tech. J.*, vol. 29, pp. 147–160; April, 1950.

locations 1, · · · , u, respectively. Here, a *memory location* shall consist of an ordered array of memory elements whose composite state represents the information stored in that array. Now suppose there are v memory elements in $M1$ which serve as the information register of an *instruction counter*, IC. Let the state of IC specify at each point which of the u instructions CL is to work on at that time. Then, v can be made the smallest integer greater than, or equal to $\log_2 u$, and successive count representations in IC can be made to differ in only one binary place. Finally, let all *data words* be stored in memory locations of $M1$. A data word is considered to be a coded problem variable which is to be operated on by C.

Let there be the same number of binary digits in data words as in instructions, and label this number y. Then assume that there are p memory elements in $M2$ which serve as the information register of a *digit counter*, DC, where p is the smallest integer greater than, or equal to, $\log_2 y$. Let DC's successive count states differ in only one binary place, and arrange CL so that if at any time DC's state represents the number j, where $1 \leqslant j \leqslant y$, CL will attempt to calculate the jth bit of the data word currently being worked on.

Now, suppose that the starting mode of C has just been completed; *i.e.*, the instructions and data have all just been stored in M. Suppose further that the states of IC and DC at this point represent the count 1, and that the state of M^* is 0. Finally, suppose that each instruction stored in $M1$ is coded up in the form: $f, A, \cdots, B, D$, where this representation means "perform operation f on the variable values stored at memory locations $A, \cdots, B$ in $M1$, and store the result at memory location D in $M1$."

Then, C can be designed to operate as follows: IC's count state of 1 causes CL to start executing the first instruction, for example $g, E, \cdots, H, L$. DC's 1 count specifies that CL is to calculate the first digit of $g(x_E, \cdots, x_H)$, where the x_i are the variable values in locations i, and place it in the first digit position at memory location D. Now in accomplishing this, CL must also set M^*'s state to 1. After M^* is set to 1, CL increases the count state of DC in $M2$ by 1 and resets M^* to 0. Next, CL calculates the second digit of $g(x_E, \cdots, x_H)$, and enters it into the second digit position at D, setting M^* to 1 again. After this, CL counts up one more in DC. Then CL sends $g(x_E, \cdots, x_H)$'s third digit into the third digit position at D. This pattern of operation continues until the yth digit of $g(x_E, \cdots, x_H)$'s value is sent into the yth digit position at D. At this point, the count state of DC is returned to 0 [if $p \neq \log_2 (y+1)$, this must be done in successive steps; *i.e.*, CL must alternately count up 1 in DC and reset M^* to 0, and set M^* back to 1 without changing the state of $M1$]. Then, the count state of IC is increased to 1. After this, CL increases the count state of DC to 1, and C is thereby prepared to start executing the second instruction.

C runs through the remaining u-1 instructions in the same manner as described above, and may complete the required computation by finally putting out all results in sequence over the O_j output bundles.

As a final note, it can easily be demonstrated that no special difficulties are presented if the above scheme of operation includes the execution of instructions which call for either a change in another instruction or a change of the count in the instruction counter. Hence, C might very easily be designed to include various types of conditional and unconditional transfer instructions, as well as other usual types of logical and arithmetic instructions.

Conclusions

One of the principal lessons of this paper is that it is in fact possible to trade equipment-domain redundancy for time-domain redundancy in digital computers whose components never fail permanently. This is demonstrated in the case of C. C has the property that while it is error-correcting over-all, it has only an error-detecting combinational part. The price paid for the corresponding reduction in equipment is that C's computation rate is slowed down. On the average, C is effectively idle for an amount of time which depends upon the probabilities of failure of its components. The exchange of equipment for time is realized by designing C so that every time a change occurs in its operating state, this change is a correct one—provided no more than an allowed number, k, of component failures occur at the same time. The essential design requirement is that C must have memory elements whose state cannot be mistakenly changed by fewer than k component failures per time interval.

Admittedly, C's failure characteristics are highly idealized. Nevertheless, in situations where transient-type failures among active network elements are the greatest concern, it would seem entirely practical to try to develop an error-correcting computer based on C's general scheme of operation. Perhaps the greatest application of such a computer would be as a super-reliable checking device to compare the outputs of other computers.

References

[1] W. E. Dickinson and R. M. Walker, "Reliability improvement by the use of multiple element switching circuits," *IBM J. Res. & Dev.*, pp. 142–147; April, 1958.

[2] P. Elias, "Computation in the presence of noise," *IBM J. Res. & Dev.*, pp. 346–353; October, 1958.

[3] B. J. Flehinger, "Reliability improvement through redundancy at various systems levels," *IBM J. Res. & Dev.*, pp. 148–158; April, 1958.

[4] H. Garner, "Generalized parity checking," IRE Trans. on Electronic Computers, vol. EC-7, pp. 207–213; September, 1958.

[5] W. Kilmer, "The Reliability Problem in Digital Computer Nets," Ph.D. dissertation, University of Michigan, Ann Arbor, Mich.; September, 1958.

[6] L. Läfgren, "Automata of high complexity and methods of increasing their reliability by redundancy," *Information and Control*, vol. 1, pp. 127–147; May, 1958.

[7] E. F. Moore and C. E. Shannon, "Reliable circuits using less reliable relays," *J. Franklin Inst.*, parts. I and II, pp. 191–208, 281–297; September/October, 1958.

[8] W. W. Peterson, "On checking an adder," *IBM J. Res. & Dev.*, pp. 166–168; April, 1958.

[9] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in "Automata Studies," C. E. Shannon and J. McCarthy, Eds., Princeton University Press, Princeton, N. J., pp. 43–98; 1956.

System Organization of a Multiple-Cockpit Digital Operational Flight Trainer*

H. J. GRAY, JR.†, H. H. NISHINO‡, AND A. L. VIVATSON†

Summary—This paper describes the system organization of a digital computer whose purpose is to activate simultaneously more than one cockpit of an operational flight trainer. The simulated aircraft are assumed to be all of the same type, but each is simulated independently. The computer is drum-sequenced and represents an application of the theory of multiple computers, since there are several different kinds of memories and more than one arithmetic unit in the system.

Introduction

IN 1950, the Moore School of Electrical Engineering was requested by the (then) Special Devices Center to investigate the feasibility of the activation of the cockpit of an operational flight trainer by a digital computer. Previous operational flight trainers had been activated by analog computers. Study showed that digital computer activation was possible and had several advantages, foremost of which was the flexibility possessed by a digital computer. This led to the design of the UDOFT[1,2] (Universal Digital Operational Flight Trainer) which is now being built by Sylvania Electric Products, Inc. In the course of the design studies, the requirements that a digital OFT had to meet were determined; about 50 msec was found to be a sufficiently fast computing cycle and parallel computing techniques were found acceptable. Recently, the Moore School was asked to see if any saving in equipment would accrue if the multiple computer design techniques being developed under another contract[3] were applied to the problem of actuating more than one cockpit; the answer to this question was definitely in the affirmative. The design that evolved is described in the following sections. The chief saving in equipment has been the replacement of an expensive fast-access storage device for the instructions for each cockpit computer by a single drum-storage unit which is shared by all the cockpit computers. This paper is a report of a study and, as yet, no equipment has been built.

* Manuscript received by the PGEC, February 11, 1959; revised manuscript received, June 10, 1959. This work was done as part of a project sponsored by the U. S. Naval Training Devices Center, Port Washington, N. Y., under contract No. N61339-272.

† The Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Pa.

‡ Radio Corporation of America, Moorestown, N. J.; formerly with The Moore School of Electrical Engineering.

[1] W. H. Dunn, C. Eldert, and P. V. Levonian, "A digital computer for use in an operational flight trainer," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-4, pp. 55–63; June, 1955.

[2] H. J. Gray, Jr., "Digital computer solution of differential equations in real time," *Proc. WJCC*, pp. 87–91; March, 1959.

[3] "Theory of Switching Circuits," Dept. of the Army, Contract DA-36-039-sc-72344 sponsored by the U. S. Army Signal Supply Agency, Lab. Procurement Support Office, Fort Monmouth, N. J.

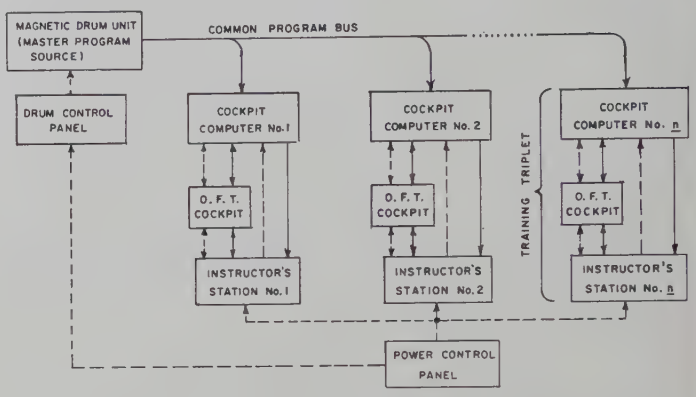


Fig. 1—Block diagram of the multiple OFT computer system.

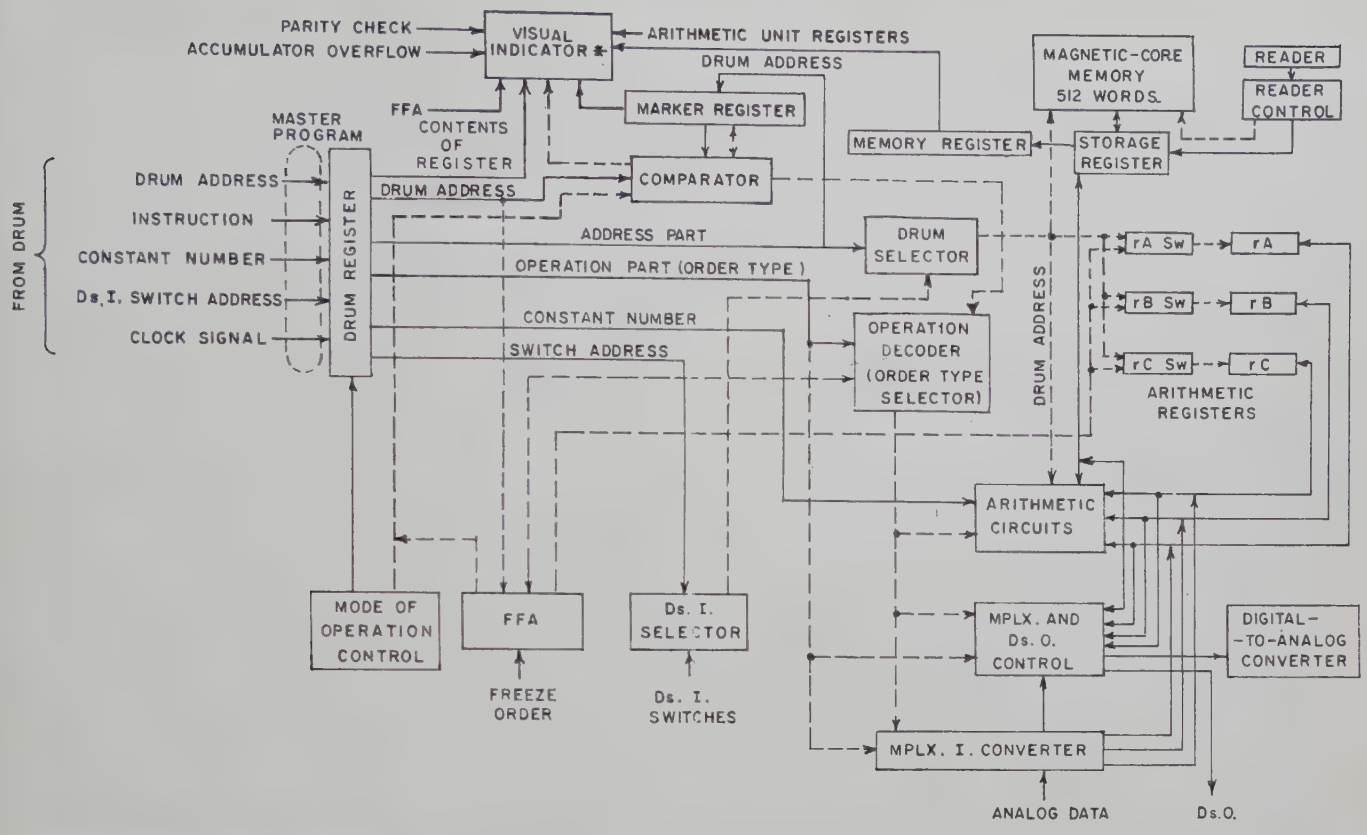
System Description

The multiple-cockpit digital operational flight trainer is a system involving control of several processes of identical type by a stored master program. Each training cockpit position has a cockpit computer and an instructor's console associated with it, the set of three units being called the training triplet. Fig. 1 shows the direct relationships in the control signal channels among the units in the triplet. A cockpit computer responds to input data from its associated cockpit and produces control signals for actuating the cockpit in real time in accordance with the master program. A minimum facility requires a drum unit and one training triplet, with facility expansion effected by increasing the number of training triplets.

A separation is made of data common to all training triplets, such as the characteristics of the aircraft being simulated, and of other data peculiar to the status of the individual training triplets. The result is an economical storage arrangement consisting of: 1) a master program drum containing all of the instructions, constants, and synchronizing signals for system-wide use, providing space for 8192 words; and 2) individual 5-μsec-access magnetic-core memory units in each cockpit computer for 512 variable words. The size of these storage media has been chosen to be larger than the present requirements to allow for simulating faster aircraft which require a larger number of instructions in the calling sequence and more data.

The magnetic drum has been inscribed prior to insertion into this machine. No alterations on the information stored on the drum are allowed under program control in this machine.

The cockpit computers are of the parallel-logic, binary-language type using 20-bit precision on data



* *Note:* Additional visual indications are provided for the contents of the following components connected via manual selector switches: rA, rB, rC, Divisor Register, Dividend Register, and Marker Register.

Fig. 2—Block diagram of a typical cockpit computer.

with one extra bit each for algebraic sign and parity. The computers receive orders from the instructor's console to which the computation responds accordingly, depending upon whether the order is a freeze order, a change of operating mode, or a discrete-input switch signal. Each cockpit computer may be started with any applicable set of initial conditions loaded in its core memory for the program at hand, and each cockpit may be given individual attention by the instructor and may be subjected to unique situations with no crosstalk between cockpits arising in the computer circuits. Complete isolation exists between cockpit computers because the flow of shared information is in one direction only, from the magnetic drum to the several cockpit computers. The block diagram of a typical cockpit computer is shown in Fig. 2.

The computing units "listen in" on the running program, which is repeated at least every 50 msec, and utilize only the applicable portions. In an effort to simplify the machine synthesis, no provision has been made for loop control in the program in the design of this computer. Thus, conventional repeatable subroutines are not used; every instruction that is to be executed is written in the program, wherever needed. Branching and re-entry in the program are handled by a conditional transfer-of-control operation. If no transfer of control is called for, the cockpit computer responds to the next instruction coming from the drum. If a transfer of control is required, the location of the instruction to which the transfer of control is to be made is placed in the marker register, and processing ceases until that instruction is reached in the running master program.

The drum has a clock-pulse channel for synchronizing the system. The elements of the computer are asynchronous in operation. These elements, when combined into the units of the computer, are synchronized by the timing signal which marks the start of each interval of time; the intervals are not necessarily equal. (The nonuniformity of the intervals arises from the inability to hold a stringent tolerance on the spacings between the timing marks on the drum.)

Drum Unit Characteristics

Loading of the drum is done by auxiliary equipment, using the reading heads for writing. The loader accepts cards or tapes on which final coding for this machine has been prepared through a keyboard, or compiled on a general-purpose computer. Changing the aircraft type under simulation requires the substitution of a new drum program.

The approximate dimensions for a drum with 8192 instructions, 72 bits in word length, are one foot in diameter (assuming a packing density per track of 200 bits per inch) and 14 inches in axial length (assuming 72 bits on parallel tracks, with 0.2-inch separation be-

tween track centers). The estimated speed is 1200 rpm for 50-msec cyclic program intervals, making the clock rate 200 kc.

The drum word is composed of 13 bits for the drum address, 1 bit for parity information, and 58 bits for the instruction as follows: 4 bits for the operation part (order type), 10 bits for the first address A_1, 14 bits for the second address A_2, 3 bits for the third address A_3, 21 bits for the number part including sign, and 6 bits for the discrete-input switch address.

Description of the Cockpit Computer

The manually controlled modes of operation of each cockpit computer are as follows.

1) Normal mode. The computer is responsive to all the instructions in the master program.
2) Freeze-on mode. During freeze-on, initiated either manually or automatically, computation is suspended except for analog output information. Thus, only the form 2 multiplex-out order (see instruction code for description) is observed, since FFA (flip-flop A—defined in the text that follows) is set. This mode is described in more detail below.
3) Halt mode. Computer is activated (secondary power is on) but is in standby. The entire master program is ignored.
4) One-instruction mode. This allows manual execution of one instruction at a time. The computer remains in the freeze-on mode (FFA is set) until the execute-instruction button is depressed, whereupon FFA is reset and the next instruction in the calling sequence is carried out. The machine reverts to the freeze-on condition (FFA set) at the end of that instruction.

The marker register participates in the above modes of operation and also makes possible the operations peculiar to the means of programming adopted for the system. In general, the marker register fills a role somewhat similar to the control counter (or instruction location counter) in a three-address coded machine of the conventional type. However, the function performed is still greater, as is explained below.

When the marker register (rM) is cleared, the operation decoder or order-type selector responds to each successive instruction read from the drum to the common program bus. When an address is stored in the marker register, the operation decoder is able to respond only to the instruction stored at that address. For example, at the start of computation, the location of the first instruction in the cyclic master program is placed into the marker register. This action transfers control to that instruction when the computer is taken out of freeze. Freeze-off may be ordered at any arbitrary position of the drum, but the computation does not start until the "marked" instruction in the master program is transmitted from the drum, upon which rM is cleared, allowing operations to begin. At any time after that, whenever the freeze-on mode is entered, the drum address of the first drum instruction is put into the marker register.

When branching is called for, the address of the instruction which is to succeed the present one is put into the marker register, and until that marked instruction is encountered, only the multiplex-out instructions are observed by the operation decoder. Meanwhile, the master program flows by; when the marked instruction is recognized by the comparator, the marker register is cleared, enabling that instruction and all those that follow to be executed as long as the marker register remains cleared. The multiplex-out instruction is an exception to all the others in that its execution and its action are independent of the marker register. The operation decoder ignores the loaded marker register on the multiplex-out form 2 command and, in turn, the register is not cleared by that instance command. The comparator is the only device which may clear that register.

During the execution of a nonbranching instruction in the one-instruction mode, the contents of the marker register are increased by unity in order that successive addresses will be carried, as in the case of the conventional control counter. Also, as in the case of the control counter, the count is replaced by the address to which control is being transferred when branching is ordered. In this one-instruction mode, although the function of the comparator is to recognize the marked instruction after the execute button is depressed, it does not clear the marker register; nevertheless, the operation decoder is allowed to respond to that instruction. Entrance to the one-instruction mode and exit from the halt mode are made through the freeze mode so that the marker register will be initially loaded with the first drum address.

The block marked FFA is a toggle device which, when set as a result of the freeze-on order from the instructor's position, modifies the action of the marker-register-and-comparator team and the operation decoder so that only the form 2 multiplex-out instruction is executed. It also prevents the clearing of the marker register as long as FFA is set. FFA may be set only after the passing of the final instruction and just before the first instruction arrives in the cyclic master program after receiving the freeze-on order. Upon removal of the freeze order, FFA may be reset immediately without the risk of starting at the wrong point in the master program, since the marker register is loaded with the first drum address.

When the drum information refers to a discrete-input controlled quantity, the corresponding switch address, which accompanies the instruction under consideration, is sent to the discrete-input selector. A check is made with the designated switch to determine whether it is ON or OFF. If, and only if, it is ON, a pulse is sent to the drum selector which blocks the associated number from the arithmetic unit. The blocking is accomplished by nullifying the address part, in effect calling for the *null register* whose address is zero. The null register does not physically exist; thus, reference to it results in obtaining binary zero as information received.

The reader is an off-line device for entering initial numerical data into the memory. It may be a reader of punched cards, paper or magnetic tape, or other convenient media carrying not only the data within the memory, but the respective addresses to which the numbers are to be sent.

The operation decoder, or order-type selector, is a device for interpreting the operation part of the instruction code and for providing control signals to the arithmetic unit. Its functional description is explained in the section on the instruction code. Its action is subject to modification by the signals from the comparator and marker register, and by FFA.

The arithmetic unit consists of an algebraic adder, arithmetic registers rA, rB, and rC, shift counter, divisor and dividend registers, overflow indicator, and a parity-checking circuit. It receives stored numerical data from the drum selector and from the magnetic-core memory. It is able to put data into the core memory when executing the read-write command; otherwise, it may hold data within the arithmetic registers. It may also receive or furnish data from or to the outside of the computer through analog and discrete input-output channels.

The visual indicator has provision for inspecting the contents of the registers in the arithmetic unit, the marker register, drum register, and storage register and for inspecting other items such as the parity check, accumulator overflow, and freeze status. All are monitored continuously except for the following which, one at a time, are selected by a manual switch: rA, rB, rC, divisor register, dividend register, and the marker register.

The drum register is connected to the common program bus to receive the entire word transmitted by the drum. Upon receipt of a new word, its contents are erased and are replaced by the more recent data. In the one-instruction mode of computer operation, only the marked (*i.e.*, current) instruction word is viewed in the visual indicator.

Instruction Code

Fifteen of the sixteen instructions provided for have been formulated. They are:

1) Add (A).
2) Absolute Add (AA).
3) Subtract (S).
4) Absolute Subtract (AS).
5) Multiply (M).
6) Multiply-Add (MA).
7) Divide (D).
8) Shift-Add (SHA).
9) Shift (SH).
10) Read-Write (RW).
11) Compare (C).
12) Multiplex In (MLXI).
13) Multiplex Out (MLXO).
14) Discrete Out (DSO).
15) Skip (SK).

The first eight are arithmetic instructions, and the last seven are control and input-output instructions.

The instruction code is the three-address type (operand No. 1, operand No. 2, and destination of the result). Certain restrictions must be observed when using these instructions.

1) A_3 may refer only to the arithmetic registers or the null register.
2) A_1 and A_2 may not both refer to magnetic-core memory locations during the same reference cycle unless they refer to the same location.

On the following description, whenever there may be ambiguity as to whether the address or the contents of a computer position are being referred to, parentheses ( ) will be used to mean "contents of;" *e.g.*, (A) means the contents of register A. Parentheses will be used only when necessary.

Description of Instructions

1) A: $(A_1)+(A_2)\rightarrow A_3$
2) AA: $(A_1)+|(A_2)|\rightarrow A_3$
3) S: $(A_1)-(A_2)\rightarrow A_3$
4) AS: $(A_1)-|(A_2)|\rightarrow A_3$
5) M: $(A_1)\cdot(A_2)\rightarrow A_3$
6) MA: $(A_1)\cdot(A_2)+(A_3)\rightarrow A_3$
7) D: $(A_1)\div(A_2)\rightarrow A_3$
8) SHA: Shift (A_2) right or left as specified by A_1, and then $(A_2)_{\text{shifted}}+(A_3)\rightarrow A_3$.
   Overflow on shift left will cause (A_2) to become the maximum number (in absolute value) that can be carried by the computer.
9) SH: Shift (A_2) right or left, as specified by A_1, and then $(A_2)_{\text{shifted}}\rightarrow A_3$.
   Overflow on shift either right or left will have no effect on the remaining bits. SH is to be used when it is desired to extract certain bits from a word.
10) RW: $(A_2)\rightarrow A_3$
    $(A_1)\rightarrow A_2$
11) C: If $(A_1)\geq 0$, go to the next instruction.
    If $(A_1)<0$, $A_2\rightarrow rM$, and computations cease until drum address $=(rM)$. C can be used as an unconditional jump by storing a negative number as the (A_1) examined by the C instruction and the desired next address as A_2.
12) MLXI: $AI_i\rightarrow(A_2)$.
    $AI_i\rightarrow(A_3)$.
    A_1 specifies the desired analog input.
13) MLXO:
    Form 1: $(A_3)\rightarrow AO_i$.
    $(A_3)\rightarrow A_2$.
    Form 2: $(A_2)\rightarrow AO_i$.
    For both forms, A_1 specifies the proper analog output channel, and A_2 can only be a core memory location. The condition specified by the Freeze switch de-

termines which MXLO form is to be used at any time, and the programmer has no control over this.

14) DSO: If $(A_2) < 0$, $1 \rightarrow DO_i$.
If $(A_2) \geq 0$, $0 \rightarrow DO_i$.
A_1 specifies the proper discrete output, and A_2 refers only to the arithmetic registers.

15) SK: Start nothing new and ignore the address part of the word. SK will be used whenever it is desirable to send nothing new to the computer, such as after instructions which require more than one word cycle for execution; or to fill the drum when the program does not require the full capacity of the drum.

There are two switch-controlled operations which will be considered here because of their close relation to the instructions. They are Discrete Input and Freeze.

DSI: The Discrete Input operation provides that the (A_2) is to be used if the associated DI switch is OFF. If the switch is ON, "zero" is used instead of (A_2).

Freeze: The Freeze operation which has been described earlier is provided to stop the computations in mid-flight for a short time (so that the instructor can talk with the student pilot or make certain checks) and then to resume computations from the same point. During this "cease computation" period, however, the computer must continue to multiplex out the analog outputs so that the instruments will continue to indicate to indicate the proper quantities.

Programming Considerations

A study was made to determine how Multiple-UDOFT with three-address code will compare with the one-address code UDOFT. For this study, it was assumed that the Multiple-UDOFT will have a minor cycle equal to that of UDOFT, and that the new instructions will require the same number of minor cycles for execution as do the instructions for UDOFT.

It was found that there will be no saving of time with the three-address code for simple sums of variables because only one variable can be obtained from the magnetic-core memory during a given reference cycle. The three-address computer will be at least as fast as UDOFT for polynomial evaluations. For sums of squared variables and for O_{33} quadrature formula evaluations, it provides a saving on the order of 30 per cent of the time required by one-address UDOFT; and for updating a set of three time-different values of the same variable, the saving is on the order of 25–40 per cent. These comparisons were made on the basis of a UDOFT without subroutines, since the Multiple-UDOFT cannot have a subroutine structure (because of the drum).

Under true UDOFT conditions, subroutines would be used. This represents a considerable saving in programming and instruction storage space at the expense of computation time. Thus, from the more realistic comparison of Multiple-UDOFT without subroutines vs UDOFT with subroutines, if was found that Multiple-UDOFT saves on the order of 15 per cent of the time required by UDOFT for a typical "flight" path computation.

The CORDIC Trigonometric Computing Technique*

JACK E. VOLDER†

***Summary*—The *CO*ordinate *R*otation *DI*gital *C*omputer (CORDIC) is a special-purpose digital computer for real-time airborne computation. In this computer, a unique computing technique is employed which is especially suitable for solving the trigonometric relationships involved in plane coordinate rotation and conversion from rectangular to polar coordinates. CORDIC is an entire-transfer computer; it contains a special serial arithmetic unit consisting of three shift registers, three adder-subtractors, and special interconnections. By use of a prescribed sequence of conditional additions or subtractions, the CORDIC arithmetic unit can be controlled to solve either set of the following equations:**

$$Y' = K(Y \cos \lambda + X \sin \lambda)$$

$$X' = K(X \cos \lambda - Y \sin \lambda),$$

or

$$R = K\sqrt{X^2 + Y^2}$$

$$\theta = \tan^{-1} Y/X,$$

where K is an invariable constant.

This special arithmetic unit is also suitable for other computations such as multiplication, division, and the conversion between binary and mixed radix number systems. However, only the trigonometric algorithms used in this computer and the instrumentation of these algorithms are discussed in this paper.

* Manuscript received by the PGEC, May 25, 1959. Presented at the Western Joint Computer Conf., San Francisco, Calif.; March 3–5, 1959.

† Convair, a Div. of General Dynamics Corp., Fort Worth, Texas.

Introduction

THE CORDIC computing technique was developed especially for use in a real-time digital computer where the majority of the computation involved the discontinuous, programmed solution of the trigonometric relationships of navigation equations and a high solution rate for the trigonometric relationships of

coordinate transformations. A prototype computer, CORDIC I, based on this computing technique, has been designed and constructed at Convair, Fort Worth. Although CORDIC I may be classified as an entire-transfer computer, its design is not based on the conventional "pencil and paper" computing technique of general-purpose computers.

Functional Description

For the sake of simplicity, the trigonometric operations in the CORDIC computer can be functionally described as the digital equivalent of an analog resolver. Similar to the operation of such a resolver, there are two computing modes, ROTATION and VECTORING. In the ROTATION mode, the coordinate components of a vector and an angle of rotation are given and the coordinate components of the original vector, after rotation through the given angle, are computed. In the second mode, VECTORING, the coordinate components of a vector are given and the magnitude and angular argument of the original vector are computed. Similarly, as in the case of resolvers, the computing device of ROTATION plus feedback is employed in the VECTORING mode. The original coordinates are rotated until the angular argument is zero, so that the total amount of rotation required is the negative of the original argument, in which case the value of the X-component is equal to the magnitude of the original vector.

In essence, the basic computing technique used in both the ROTATION and VECTORING modes in CORDIC is a step-by-step sequence of pseudo rotations which result in an over-all rotation through a given angle (ROTATION) or result in a final angular argument of zero (VECTORING).

It is necessary that the angular increments of rotation be computed in a decreasing order. There are several permissible values which may be chosen for the angular magnitude of the first rotation step. The magnitude actually chosen for the first increment is 90°. The expression for a set of coordinate components, Y_1 and X_1, rotated through plus or minus 90° is simply

$$Y_2 = \pm X_1 = R_1 \sin(\theta_1 \pm 90°) \tag{1}$$

$$X_2 = \mp Y_1 = R_1 \cos(\theta_1 \pm 90°). \tag{2}$$

The first step is unique in that a perfect rotation step is performed.

The rest of the computing steps can be clarified by examining the relationships, involved in a typical rotation step, which are shown in Fig. 1. Consider two given coordinate components, Y_i and X_i, in the plane coordinate system shown in the figure. In this discussion, the quantity i is equal to the number of the particular step under consideration. The components, Y_i and X_i, are associated with the ith step and describe a vector of magnitude R_i at an angle θ_i from the origin according to the relationship:

$$Y_i = R_i \sin \theta_i \tag{3}$$

$$X_i = R_i \cos \theta_i. \tag{4}$$

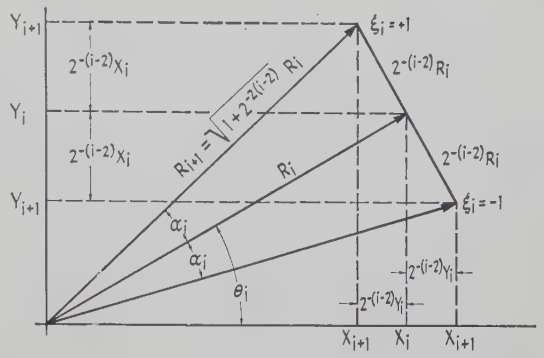


Fig. 1—Typical computing step.

In Fig. 1, the angle α_i is the special magnitude of rotation associated with each computing step. The general expression for α_i where $i>1$, is

$$\alpha_i = \tan^{-1} 2^{-(i-2)}. \tag{5}$$

The peculiar magnitude of each α_i is such that a "rotation" of coordinate components through $\pm\alpha_i$ may be accomplished by the simple process of shifting and adding.

The two choices of positive or negative rotation are shown in Fig. 1. The general expression for the rotated components is

$$\begin{aligned} Y_{i+1} &= \sqrt{1+2^{-2(i-2)}}\,R_i \sin(\theta_i \pm \alpha_i) \\ &= Y_i \pm 2^{-(i-2)}X_i \end{aligned} \tag{6}$$

$$\begin{aligned} X_{i+1} &= \sqrt{1+2^{-2(i-2)}}\,R_i \cos(\theta_i \pm \alpha_i) \\ &= X_i \mp 2^{-(i-2)}Y_i. \end{aligned} \tag{7}$$

Note that, by restricting the angular rotation magnitude to (5), the right-hand terms of (6) and (7) may be obtained by two simultaneous shift-and-add operations. This is the *fundamental* relationship upon which this computing technique is based.

The computing action of adding (or subtracting) a shifted value of X_i to Y_i to obtain Y_{i+1}, while simultaneously subtracting (or adding) a shifted value of Y_i to X_i to obtain X_{i+1}, is termed "cross-addition."

While the expressions for Y_{i+1} and X_{i+1} are not perfect rotations because of the increase in magnitude by the terms under the radical, either of the two choices of direction produces the same change in magnitude. If, therefore, for each step, the coordinates are always rotated through either a positive or negative α_i, then the increase in magnitude may be considered as a *constant*. This requirement precludes the choice of zero rotation at any step. To identify the choice made in a particular step, the $\pm$ notation may be represented by the binary operator $+\xi_i$ where ξ_i can equal either $+1$ or -1. This substitution produces the general expressions

$$\begin{aligned} Y_{i+1} &= \sqrt{1+2^{-2(i-2)}}\,R_i \sin(\theta_i + \xi_i\alpha_i) \\ &= Y_i + \xi_i 2^{-(i-2)}X_i \end{aligned} \tag{8}$$

$$\begin{aligned} X_{i+1} &= \sqrt{1+2^{-2(i-2)}}\,R_i \cos(\theta_i + \xi_i\alpha_i) \\ &= X_i - \xi_i 2^{-(i-2)}Y_i \end{aligned} \tag{9}$$

where

$$\xi_i = +1 \text{ or } -1. \tag{10}$$

Likewise, after the completion of the rotation step in which the $i+1$ terms are obtained, the $i+2$ terms may be computed from these terms with the results

$$Y_{i+2} = \sqrt{1+2^{-2(i-1)}}\sqrt{1+2^{-2(i-2)}}R_i \cdot \sin(\theta_i + \xi\alpha_i + \xi_{i+1}\alpha_{i+1}) \tag{11}$$

$$X_{i+2} = \sqrt{1+2^{-2(i-1)}}\sqrt{1+2^{-2(i-2)}}R_i \cdot \cos(\theta_i + \xi_i\alpha_i + \xi_{i+1}\alpha_{i+1}). \tag{12}$$

Similarly, the pseudo-rotation steps can be continued through any finite, pre-established number of steps without regard to the values assigned to ξ. Consider the initial coordinate components Y_1 and X_1 where

$$Y_1 = R_1 \sin\theta_1 \tag{13}$$

$$X_1 = R_1 \cos\theta_1. \tag{14}$$

By establishing the first and most significant step as a rotation through $\pm 90°$, and by establishing the number of steps as n, the expression for the final coordinate components will be

$$Y_{n+1} = [\sqrt{1+2^{-0}}\sqrt{1+2^{-2}}\cdots\sqrt{1+2^{-2(n-2)}}]R_1 \cdot \sin(\theta_1 + \xi_1\alpha_1 + \xi_2\alpha_2 + \cdots + \xi_n\alpha_n) \tag{15}$$

$$X_{n+1} = [\sqrt{1+2^{-0}}\sqrt{1+2^{-2}}\cdots\sqrt{1+2^{-2(n-2)}}]R_1 \cdot \cos(\theta_1 + \xi_1\alpha_1 + \xi_2\alpha_2 + \cdots + \xi_n\alpha_n) \tag{16}$$

The increase in magnitude of the components for a particular value of n is a constant and will be represented by the letter K. The value selected for n is a function of the desired computing accuracy and can be a constant for a particular computer. For example,

$$\text{if } n = 24, \quad K = 1.646760255. \tag{17}$$

The necessary functional components and information flow for instrumenting the cross-addition are associated with the Y and X registers shown in Fig. 2.

It has not yet been shown how the prescribed sequence of rotation steps can be controlled to effect the desired over-all rotation. By examination of (15) and (16), it may be shown that, for a rotation of a set of coordinate components Y_1 and X_1 through a given angle (as required in the ROTATION mode), it is necessary to obtain the expressions

$$Y_{n+1} = KR_1 \sin(\theta_1 + \lambda) \tag{18}$$

$$X_{n+1} = KR_1 \cos(\theta_1 + \lambda). \tag{19}$$

To obtain the relationships expressed in (18) and (19), it is required that

$$\lambda = \xi_1\alpha_1 + \xi_2\alpha_2 + \cdots + \xi_n\alpha_n \tag{20}$$

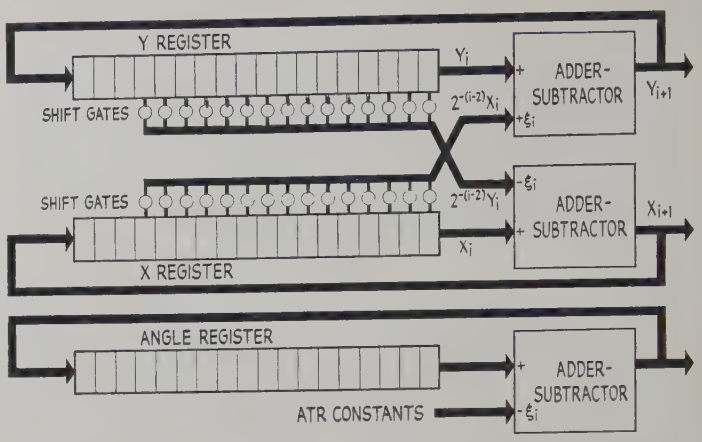


Fig. 2—CORDIC arithmetic unit.

and, as explained previously, for VECTORING it is required that

$$-\theta_1 = \xi_1\alpha_1 + \xi_2\alpha_2 + \cdots + \xi_n\alpha_n. \tag{21}$$

The sequences of (20) and (21) form a special radix representation equivalent to the desired angle, λ or θ where

$$\alpha_1 = 90° \tag{22}$$

$$\alpha_2 = \tan^{-1} 2^{-0} = 45° \tag{23}$$

$$\alpha_3 = \tan^{-1} 2^{-1} \approx 26.5° \tag{24}$$

$$\alpha_i = \tan^{-1} 2^{-(i-2)}. \tag{25}$$

The α terms are referred to as ATR (Arc Tangent Radix) constants, and are precomputed and stored in the computer. The ξ terms are referred to as ATR digits and are determined during each operation.

In the CORDIC computer, the ATR digits are determined sequentially, most significant digit first, and are used to control the conditional action of the adder-subtractors in the arithmetic unit. The following paragraphs contain a description of the manner in which the ATR code representation, $\xi_1\xi_2\xi_3\cdots\xi_n$, can be determined for any given angle, λ or θ.

First, for any angle, λ or θ, there must be at least one set of values for the ξ operators that will satisfy (20) or (21). Second, a simple technique must be available for determining the ATR code digits that satisfy these equations.

The following relationships are necessary and sufficient for *any* sequence of radix constants to meet the above requirements.

$$|\lambda \text{ or } \theta| \leq \alpha_1 + \alpha_2 + \alpha_3 + \cdots + \alpha_n + \alpha_n \tag{26}$$

$$\alpha_i \leq \alpha_{i+1} + \alpha_{i+2} + \cdots + \alpha_n + \alpha_n. \tag{27}$$

For the satisfaction of the stipulative equations [(10) and (22)], it is required that λ or θ be represented

$$-180° \leq \lambda \text{ or } \theta < +180°. \tag{28}$$

Eq. 28 imposes no special consideration if the two's complement notation is used.

By employing an additional register and adder-subtractor (identified in Fig. 2 as the angle register) the relationship of (18) (ROTATION mode) can be instrumented by: 1) sensing the sign of the angle of ROTATION (or remainder if $i>1$); and 2) either subtracting or adding to the angle the ATR constant corresponding to the particular step. In each step, the relationship instrumented is:

$$|\lambda_{i+1}| = \big|\,|\lambda_i| - \alpha_i\big|. \tag{29}$$

Execution of the first step of the nulling sequence to (26) results in

$$-\alpha_1 \le |\lambda| \overset{\bullet}{-} \alpha_1 \le \alpha_2 + \alpha_3 + \cdots + \alpha_n + \alpha_n. \tag{30}$$

Application of the relationships of (27) results in

$$|\lambda_2| \equiv \big|\,|\lambda_1| - \alpha_1\big| \le \alpha_2 + \alpha_3 + \cdots + \alpha_n + \alpha_n. \tag{31}$$

Continuation of the nulling sequence through α_n results in

$$|\lambda_{n+1}| \le \alpha_n. \tag{32}$$

Eq. (32) can be used to prove that the remainder in the angle register converges to zero in the ROTATION mode.

The sequence of operation signs used to null λ to zero is the negative of the equivalent ATR code for the original angle λ_1. More simply, the ATR code digit of each step is equal to the sign of the quantity in the angle register before each step. Therefore, simultaneously with each nulling step in the angle register, the ATR code digit may be used to control the cross-addition step in the Y and X registers (shown in Fig. 2) to effect a rotation of components through an equal angular increment.

The proof of the convergence of the effective angular argument θ_{n+1} to zero, which is necessary in the VECTORING mode, may be obtained by replacing λ by θ in (29) through (32). In VECTORING, the sign of the angle θ_i is obtained by sensing the sign of Y_i. The sequence of signs of Y_i is the negative of the ATR code for the effective rotation performed on the components Y_1 and X_1. During each cross-addition operation in the Y and X register, the corresponding ATR constant can be conditionally added or subtracted, depending on ξ_i, to an accumulating sum in the angle register so that, at the end of the computing sequence, when $\theta_{n+1}=0$, the quantity in the angle register will be equal to the original angular argument θ_1 of the coordinate components Y_1 and X_1.

The step-by-step results of a typical rotation computing sequence are shown in Table I. The two's complement notation is used for all quantities, and, for simplicity, shifted quantities are simply truncated without round-off.

The step-by-step results of a typical vectoring computing sequence are shown in Table II.

Fig. 3 contains the solution flow for solving a typical navigation problem with the CORDIC computing technique. Specifically, this example shows the program sequence necessary for solving for the course angle and the distance-to-destination relationships in great circle steering.

TABLE I

TYPICAL ROTATION COMPUTING SEQUENCE

Y Register	X Register	Angle Register	
$Y_1=0.0101110$	$1.1000101=X_1$	$0.1100101=\lambda$	
$+1.1000101$	-0.0101110	-0.1000000	$\tan^{-1}\infty$
1.1000101	1.1010010	0.0100101	
$+1.1010010$	-1.1000101	-0.0100000	$\tan^{-1}1$
1.0010111	0.0001101	0.0000101	
$+0.0000110$	-1.1001011	-0.0010010	$\tan^{-1}2^{-1}$
1.0011101	0.1000010	1.1110011	
-0.0010000	$+1.1100111$	$+0.0001001$	$\tan^{-1}2^{-2}$
1.0001101	0.0101001	1.1111100	
-0.0000101	$+1.1110001$	$+0.0000101$	$\tan^{-1}2^{-3}$
1.0001000	0.0011010	0.0000001	
$+0.0000001$	-1.1111000	-0.0000010	$\tan^{-1}2^{-4}$
1.0001001	0.0100010	1.1111111	
-0.0000001	$+1.1111100$	$+0.0000001$	$\tan^{-1}2^{-5}$
1.0001000	0.0011110	0.0000000	

TABLE II

TYPICAL VECTORING COMPUTING SEQUENCE

Y Register	X Register	Angle Register	
$Y_1=0.0101110$	$1.1000101=X_1$	0.0000000	
-1.1000101	$+0.0101110$	$+0.1000000$	$\tan^{-1}\infty$
0.0111011	0.0101110	0.1000000	
-0.0101110	$+0.0111011$	$+0.0100000$	$\tan^{-1}1$
0.0001101	0.1101001	0.1100000	
-0.0110100	$+0.0000110$	$+0.0010010$	$\tan^{-1}2^{-1}$
1.1011001	0.1101111	0.1110010	
$+0.0011011$	-1.1110110	-0.0001001	$\tan^{-1}2^{-2}$
1.1110100	0.1111001	0.1101001	
$+0.0001111$	-1.1111110	-0.0000101	$\tan^{-1}2^{-3}$
0.0000011	0.1111011	0.1100100	
-0.0000111	$+0.0000000$	$+0.0000010$	$\tan^{-1}2^{-4}$
1.1111100	0.1111011	0.1100110	
$+0.0000011$	-1.1111111	-0.0000001	$\tan^{-1}2^{-5}$
1.1111111	$0.1111100=KR_1$	$0.1100101=\theta$	

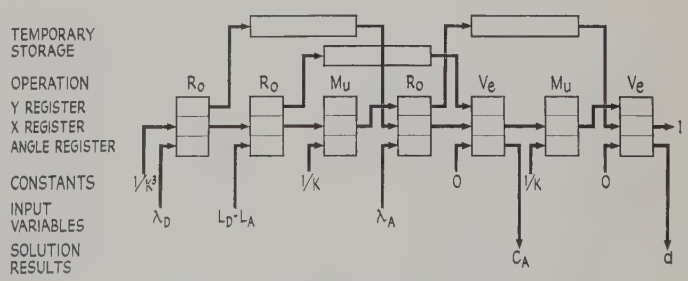


Fig. 3—Solution flow diagram.

Each operation of the computing sequence is represented by a box containing the Y, X and Angle registers. The particular operation performed is abbreviated as follows:

R_0 = Rotation,

V_e = Vectoring,

M_u = Multiplication.

The explicit equations solved are

$$C_A = \tan^1 \frac{\cos \lambda_D \sin (L_D - L_A)}{\sin \lambda_D \cos \lambda_A - \cos \lambda_D \sin \lambda_A \cos (L_D - L_A)} \tag{33}$$

$$d = \tan^{-1} \frac{\cos \lambda_D \sin (L_D - L_A)}{\sin C_A [\sin \lambda_D \sin \lambda_A + \cos \lambda_D \cos \lambda_A \cos (L_D - L_A)]} \tag{34}$$

where

C_A = course angle to destination
d = distance to destination
λ_A = present latitude
L_A = present longitude
λ_D = latitude of destination
L_D = longitude of destination.

The form of (33) and (34) cannot be used for establishing the CORDIC solution-flow diagram. It is necessary to express the relationship with some form of rotation operators, such as rotation matrices; a similar change in form is also necessary for establishing analog resolver solution-flow diagrams. The only difference between Fig. 3 and an equivalent analog resolver solution flow diagram is the insertion of multiplication routines to compensate for the magnitude change factor K of each trigonometric operation. Note that, although five trigonometric operations are performed, only two multiplication operations are necessary.

Conclusion

The CORDIC computing technique is especially suitable for use in a special-purpose computer where the majority of the computations involve trigonometric relationships. In general, the ROTATION and VECTORING operations should be considered constant-length routines in which the number of word times per operation is equal to the word length.

While not covered in this paper, similar algorithms have been developed for multiplication, division, conversion between binary and mixed radix systems, extractions of square root, hyperbolic coordinate transformations, exponentiation and generation of logarithms.

It is believed that similar algorithms based on this fundamental concept of computation could be developed for many other computing requirements.

Decimal-Binary Conversions in CORDIC*

D. H. DAGGETT†

Summary—A special-purpose, binary computer called CORDIC (*CO*ordinate *R*otation *DI*gital *C*omputer) contains a unique arithmetic unit composed of three shift registers, three adder-subtractors, and suitable interconnections for efficiently performing calculations involving trigonometric functions. A technique is formulated for using the CORDIC arithmetic unit to convert between angles expressed in degrees and minutes in the 8, 4, 2, 1 code and angles expressed in binary fractions of a half revolution. Decimal-to-binary conversion is accomplished through the generation of an intermediate binary code in which the variable values are +1 and −1. Each of these intermediate code variables controls the addition or subtraction of a particular binary constant in the formation of an accumulated sum which represents the angle. Examples are presented to illustrate the technique. Binary-to-decimal conversion is accomplished by applying essentially the same conversion steps in reverse order, but this feature is not discussed fully. Fundamental principles of the conversion technique, rather than details of implementation, are emphasized. The CORDIC conversion technique is sufficiently general to be applied to decimal-binary conversion problems involving other mixed radix systems and other decimal codes.

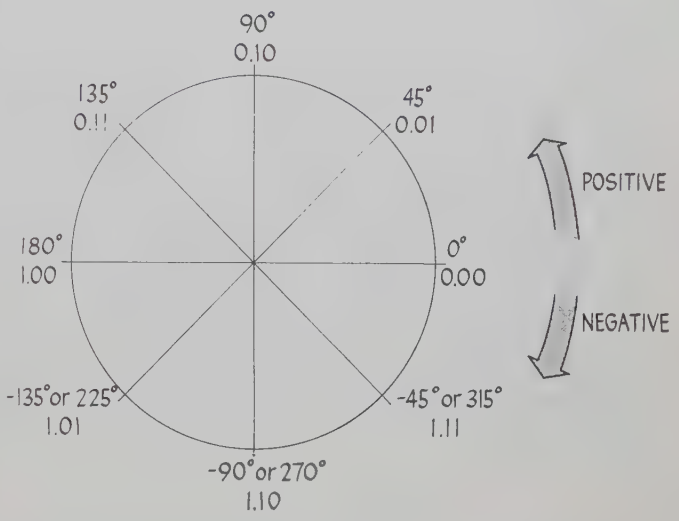


Fig. 1—Representation of angles in CORDIC.

Introduction

A unique concept for a special-purpose digital computer is described in this issue.[1] This computer embodies a unique method for the rapid serial evaluation of sines, cosines, transformations between polar and rectangular coordinates, and other operations concerned with coordinate rotations. For this reason, the name of the computer is derived from the initial letters of the words COordinate Rotation DIgital Computer.

The versatility of CORDIC is enhanced by utilizing the novel configuration of the arithmetic unit for decimal-binary conversions as well as for trigonometric calculations. Since angles are inherent in trigonometric calculations, it follows that many independent and computed variables in a specific CORDIC application will be angles. Therefore, this treatment is directed specifically to the conversion of angles.

In CORDIC, angles are represented as binary fractions of a half revolution with two's complements for negative angles, as shown in Fig. 1. Since a one to the left of the binary point is used to represent a negative quantity in the two's complement system, angles from +180° to slightly less than +360° are interpreted internally as negative angles measured clockwise from 0°.

Before the conversion technique is explained, the basic philosophy of computation in CORDIC will be summarized. the arithmetic unit, as shown in Fig. 2 of Volder's paper,[1] consists of three shift registers, three adder-subtractors, and suitable interconnections. Each programmed operation is accomplished in a fixed number of steps. Each step involves the modification of a number by adding or subtracting a constant. The settings of all three adder-subtractors are controlled by the sign of the quantity in one of the arithmetic unit registers. In this way, calculations related to the addition or subtraction of the previously mentioned constant can be executed simultaneously.

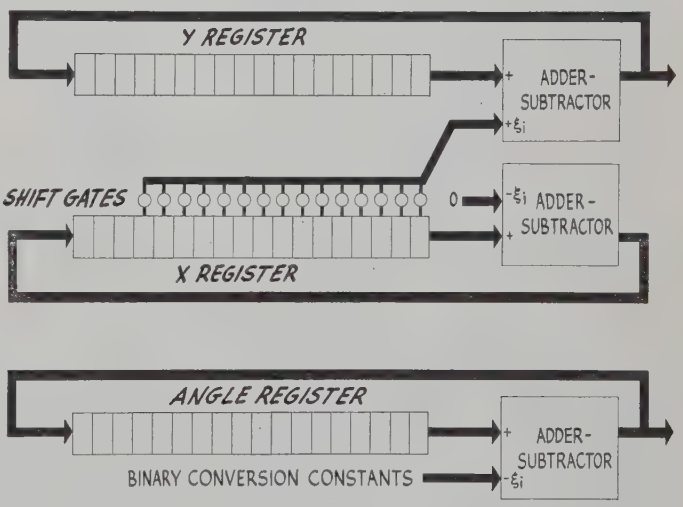


Fig. 2—Implementation of ±code to binary conversion.

Conventional Conversion Technique

The CORDIC decimal-to-binary conversion technique may be compared to a conventional conversion

* Manuscript received by the PGEC, May 22, 1959; revised manuscript received, June 25, 1959.

† Convair, a Div. of General Dynamics Corp., Fort Worth, Tex.

[1] J. E. Volder, "The CORDIC trigonometric computing technique," this issue, p. 330.

technique in which the 8, 4, 2, 1 code and binary arithmetic is utilized. The conventional conversion technique is based upon the 8, 4, 2, 1 code definition of the value of a decimal digit, N, located i places to the left of the units position, as given by

$$N \cdot 10^i = X_4(8 \cdot 10^i) + X_3(4 \cdot 10^i) + X_2(2 \cdot 10^i) + X_1(1 \cdot 10^i) \quad (1)$$

where $X_j = 0$ or 1. The constants $8 \cdot 10^i$, $4 \cdot 10^i$, $2 \cdot 10^i$, and $1 \cdot 10^i$, evaluated in binary for all values of i of be used, are required in the conversion. The transformation from degrees and minutes to fractions of a half revolution is incorporated by expressing the binary constants in the desired units. A conversion is accomplished by executing the operations indicated in (1). These operations are shown in Table I for an angle of 45°, which is expressed in 8, 4, 2, 1 code as (0100) (0101). Conversion of the magnitude of a negative angle is accomplished in the same way, and the result is then complemented by subtracting the binary magnitude from zero.

TABLE I

Constants Degrees	Constants—Binary Fraction of Half Revolution		8, 4, 2, 1 Code Variables		Product Terms
80	0.01110010	×	0	=	0.00000000
40	0.00111001	×	1	=	0.00111001
20	0.00011100	×	0	=	0.00000000
10	0.00001110	×	0	=	0.00000000
8	0.00001011	×	0	=	0.00000000
4	0.00000110	×	1	=	0.00000110
2	0.00000011	×	0	=	0.00000000
1	0.00000001	×	1	=	0.00000001
Accumulated Sum = 2^{-2} Half Revolution				=	0.01000000

Conversion Based Upon ± Code

In Table I, it is shown that at each step, a binary constant is either added or not added, depending upon whether the 8, 4, 2, 1 code variable is 1 or 0, respectively. In order to use the CORDIC principle, it is necessary either to add or to subtract a constant. The use of addition or subtraction is controlled by a code variable placed in the sign digit position of an arithmetic unit register. The problem of conversion by adding and subtracting constants will be considered first. Subsequently, the method of properly positioning the code variables for control will be presented.

By analogy to the way in which a code variable of +1 is used to establish the addition of a constant, a variable of −1 may be used to establish subtraction. Therefore, it is desired that a binary code with +1 and −1 variables be used to represent decimal angles in CORDIC. For convenience, the desired code will be called a ± (plus-minus) code. The 8, 4, 2, 1 weights cannot be applied directly to a four-digit ± code because all possible sums of binary-weighted ± code digits are odd. Therefore, a transformation of the decimal digits 0, 1, · · ·, 9 into a set of ten odd integers is necessary.

For any specific digit position, the "largest transformed value" is defined as the largest absolute value of odd integers selected for that digit position and multiplied by the appropriate power of ten. In order for the ± code to be useful in the inverse transformation from binary to decimal, it is necessary and sufficient that the following conditions be satisfied.

1) With reference to a particular digit position, the sum of the largest transformed values for all less significant digit positions must be smaller than the power of ten for the aforementioned particular digit position under consideration.
2) Condition 1 must be satisfied for each digit position.

The set of ten odd integers $-9, -7, \cdots, -1, +1, \cdots, +9$ satisfies the foregoing conditions for positions in which all decimal digits, 0 through 9, are possible, and this set will be used for illustration purposes.

The equation transforming a decimal digit N, having one of the values, 0, 1, · · · , 9, into a digit Y having, respectively, the values $-9, -7, \cdots, +9$, is

$$Y = 2N - 9. \quad (2)$$

The equation for the inverse transformation is

$$N = \tfrac{1}{2}Y + \tfrac{9}{2}. \quad (3)$$

Applying the factor of $\frac{1}{2}$ in (3) to the 8, 4, 2, 1 weights results in the ± code equation

$$N = Y_4 \cdot 4 + Y_3 \cdot 3 + Y_2 \cdot 1 + Y_1 \cdot \tfrac{1}{2} + C \quad (4)$$

where $Y_j = +1$ or -1 and $C = 9/2$. A factor of 10^i may be applied to each term in (4), as was done in (1), to account for the position of the digit N. The pattern of the Y_j variables of the code of (4), with $C = 9/2$ and with 0's used to represent -1's, is identical to that of the well-known Excess-3 code.[2]

It should be noted that the value of C in (4) is determined by the choice of integers upon which the transformation equations are based. In general, the transformation of a set of $n+1$ integers $N = 0, 1, \cdots, n$ (n odd) into odd integers $Y = -n, -n+2, \cdots, -1, +1, \cdots, n-2, n$ leads to the inverse transformation,

$$N = \tfrac{1}{2}Y + \tfrac{n}{2} \quad (5)$$

and a value of $C = n/2$. This form of transformation always leads to a ± code which (by analogy to the Excess-3 code) is equivalent to an "Excess" code. It follows that if the proper ± ("Excess") code is used for each digit position, no code conversion is necessary. However, if it is desired to use the 8, 4, 2, 1 code, the required conversion can be readily accomplished by adding a binary constant to each 8, 4, 2, 1-coded digit.

[2] R. K. Richards, "Arithmetic Operations in Digital Computers," D. Van Nostrand Co., Inc., New York, N. Y., pp. 182–184; 1955.

The basic conversion procedure illustrated for the 8, 4, 2, 1 code may be used with the ± code provided the constant term $C \cdot 10^i$ for all decimal digit positions is added in binary to the accumulated sum. As an example, 45° will be converted from ± (Excess-3) code to binary. Since the constant for this code is $(9/2)10^i$, the sum of the constants for the two digits is $49\frac{1}{2}°$. The ± code representation (with 1's omitted) for 45° is $(-+++)(+---)$, and the conversion is illustrated in Table II.

TABLE II

Constants Degrees	Constants—Binary Fraction of Half Revolution		± Code Variables		Product Terms	Accumulated Sum
$49\frac{1}{2}$	0.010001100110		(Correction)		+0.010001100110	0.010001100110
40	0.001110001110	×	−1	=	−0.001110001110	0.000011011000
20	0.000111000111	×	+1	=	+0.000111000111	0.001010011111
10	0.000011100100	×	+1	=	+0.000011100100	0.001110000011
5	0.000001110010	×	+1	=	+0.000001110010	0.001111110101
4	0.000001011011	×	+1	=	+0.000001011011	0.010001010000
2	0.000000101110	×	−1	=	−0.000000101110	0.010000100010
1	0.000000010111	×	−1	=	−0.000000010111	0.010000001011
$\frac{1}{2}$	0.000000001011	×	−1	=	−0.000000001011	0.010000000000

Accumulated Sum = 2^{-2} Half Revolution = 0.010000000000

Positioning of ± Variables for Adder-Subtractor Control

Successive digits of the ± code must control successive settings of the adder-subtractors in order for the proper sequence of additions and subtractions to occur as indicated in Table II. The settings of the adder-subtractors during conversion operations are established by the value of the sign digit located in the Y Register (decimal-to-binary) or in the Angle Register (binary-to-decimal). The settings relative to each other are indicated in Fig. 2 by the sign of ξ_i, a +1 or −1 variable determined by the value of the controlling sign digit.

In positioning the ± code digits for control, the technique of binary nonrestoring division is useful because in this operation, successive quotient digits are given by the signs of successive remainders. Dividing the number representing the ± code of the angle by 1 produces the complements of successive ± code digits (in space) as the signs of successive remainders (in time). This binary nonrestoring division by 1 is easily implemented in CORDIC by means of the customary rules given below.

1) If remainder is positive, subtract divisor; if remainder is negative, add divisor.
2) Shift divisor one place to right.
3) Repeat 1 and 2.

The positioning of digits of the ± code for 45° is illustrated by following the above rules, as shown in Table III.

TABLE III

$(-+++)(+---)=0111$		1000	Sign of Remainder
Subtract	1		
	1111	1000	−
Add	1		
	0011	1000	+
Subtract	1		
	0001	1000	+
Subtract	1		
	0000	1000	+
Subtract		1	
	0000	0000	+
Subtract		1	
	1111	1100	−
Add		1	
	1111	1110	−
Add		1	
	1111	1111	−

The Complete Decimal-to-Binary Conversion

The positioning of ± code digits for control and the generation of the binary angle by addition and subtraction of constants take place simultaneously within the portion of the CORDIC arithmetic unit shown in Fig. 2. This simultaneous action will be illustrated by means of an example in which different ± codes from the Excess-3-equivalent code are used for certain decimal digit positions. Since the hundreds-of-degrees position requires only four digits (0–3), the application of (5) produces a constant $C = 3/2$ and a ± code which is equivalent to the 8, 4, 2, 1 (or "Excess-0") code. Only the last two binary digits of this code are required.

Similarly, a constant $C = 5/2$ is used for the tens-of-minutes position, where only six digits (0–5) are used. The corresponding ± code is equivalent to an "Excess-5" code. Three (instead of four) binary digits may be used for this position if desired.

In decimal-to-binary conversion, the ± code for the desired angle is placed in the Y register (Fig. 2), and

TABLE IV

Operation—Top Adder-Subtractor	Register Contents	Operation—Bottom Adder-Subtractor	Register Contents		Value
	$Y=0.10111010101101011$		Angle	1.00011100011011101	199°59.5′
Subtract	$X=1.0$	Add			
	$Y=1.10111010101101011$		Angle	1.00011100011011101	199°59.5′
Add	$X=0.1$	Subtract		0.10001110001110010	−100°
	$Y=0.00111010101101011$		Angle	0.10001110001101011	99°59.5′
Subtract	$X=0.01$	Add		0.01000111000111001	+ 50°
	$Y=1.11111010101101011$		Angle	0.11010101010100100	149°59.5′
Add	$X=0.001$	Subtract		0.00111000111000111	− 40°
	$Y=0.00011010101101011$		Angle	0.10011100011011101	109°59.5′
Subtract	$X=0.0001$	Add		0.00011100011100100	+ 20°
	$Y=0.00001010101101011$		Angle	0.10111000111000001	129°59.5′
Subtract	$X=0.00001$	Add		0.00001110001110010	+ 10°
	$Y=0.00000010101101011$		Angle	0.11000111000110011	139°59.5′
Subtract	$X=0.000001$	Add		0.00000111000111001	+ 5°
	$Y=1.11111110101101011$		Angle	0.11001110001101100	144°59.5′
Add	$X=0.0000001$	Subtract		0.00000101101100001	− 4°
	$Y=0.00000000101101011$		Angle	0.11001000100001011	140°59.5′
Subtract	$X=0.00000001$	Add		0.00000010110110000	+ 2°
	$Y=1.11111111101101011$		Angle	0.11001011010111011	142°59.5′
Add	$X=0.000000001$	Subtract		0.00000001011011000	− 1°
	$Y=0.00000000001101011$		Angle	0.11001001111100011	141°59.5′
Subtract	$X=0.0000000001$	Add		0.00000000101101100	+ 30′
	$Y=1.11111111111101011$		Angle	0.11001010101001111	142°29.5′
Add	$X=0.00000000001$	Subtract		0.00000000111100101	− 40′
	$Y=0.00000000000101011$		Angle	0.11001001101101010	141°49.5′
Subtract	$X=0.000000000001$	Add		0.00000000011110011	+ 20′
	$Y=0.00000000000001011$		Angle	0.11001010001011101	142°09.5′
Subtract	$X=0.0000000000001$	Add		0.00000000001111001	+ 10′
	$Y=1.11111111111111011$		Angle	0.11001010011010110	142°19.5′
Add	$X=0.00000000000001$	Subtract		0.00000000000111101	− 5′
	$Y=0.00000000000000011$		Angle	0.11001010010011001	142°14.5′
Subtract	$X=0.000000000000001$	Add		0.00000000000110001	+ 4′
	$Y=1.11111111111111111$		Angle	0.11001010011001010	142°18.5′
Add	$X=0.0000000000000001$	Subtract		0.00000000000011000	− 2′
	$Y=0.00000000000000001$		Angle	0.11001010010110010	142°16.5′
Subtract	$X=0.00000000000000001$	Add		0.00000000000001100	+ 1′
	$Y=0.00000000000000000$		Angle	0.11001010010111110	142°17.5′
Subtract		Add		0.00000000000000110	+ .5′
			Angle	0.11001010011000100	142°18.0′

the divisor of 1 is placed in the X register. A sign digit of 0 in the Y register establishes a ξ_i of -1, which is programmed to set the top adder-subtractor (Fig. 2) to subtract and the bottom adder-subtractor to add. A sign digit of 1 has the opposite effect. The sum of constant terms designated by C in (4) is initially placed in the angle register, and successive constants are introduced to the bottom adder-subtractor as shown in Fig. 2. As one step of the division is taking place to establish the next setting of the adder-subtractors, a constant is being added or subtracted to modify the quantity in the angle register according to the sign digit in the Y register at the beginning of the step. The binary angle is taken from the bottom adder-subtractor on the final computation step.

An example of 142° 18 minutes will be converted to binary, as shown in Table IV. The correct result is 0.11001010011000100, which is identical to the result shown in Table IV.

Binary-to-Decimal Conversion

The binary-to-decimal conversion may be accomplished by a similar technique. The sum of constant terms designated by C in (4) is first subtracted from the binary angle. Then the $\pm$ code is generated by a "nonrestoring division" in which the divisors are successive constants of the $\pm$ code. Simultaneously with the "nonrestoring division," the 8, 4, 2, 1 code is formed (if desired) by adding the appropriate constant to the $\pm$ code.

The binary-to-decimal conversion is accomplished with the same constants and control logic as the decimal-to-binary conversion. The major differences occur in loading the arithmetic unit registers and control of the adder-subtractors from the angle register instead of the Y register. In this conversion, the binary angle is initially placed in the angle register, and the binary-coded-decimal angle is formed in the Y register.

Conclusion

The technique described herein was devised especially for conversions between mixed radix decimal angles and binary angles in a digital computer with a CORDIC type arithmetic unit. One of the most attractive features of the technique is that the same constants and control logic can be used in both decimal-to-binary and binary-to-decimal conversion. Although the technique was designed for a particular computer, it is felt that certain concepts will be useful in other applications. For this reason, emphasis has been placed more on fundamental concepts than on details of a specific application. Obvious modifications are possible and may be desirable in a particular application of this conversion technique.

Acknowledgment

J. E. Volder, who conceived the basic organization of CORDIC, also envisioned a conversion technique which could be accomplished in CORDIC with little or no modification of the arithmetic unit. The author is indebted to Mr. Volder for discussions and suggestions which assisted materially in the development of this conversion technique.

Minimal Sequential Machines*

DOUGLAS B. NETHERWOOD†

Summary—**The general class of sequential machines defined by Mealy is investigated. It is found that any such machine can be identified with a set of machines of equivalent minimality. A procedure for developing the aggregate of all sets of gates for such minimal machines is evolved, and the problem of selecting components for constructing machines is discussed.**

* Manuscript received by the PGEC, February 16, 1959; revised manuscript received, June 1, 1959.

† Electronic Components Lab., Wright Air Dev. Center, Wright-Patterson AFB, Dayton, Ohio.

Introduction

THIS PAPER describes a procedure for commencing with the *transition diagram* of a sequential machine such as that of Fig. 1 and deriving a minimal equivalent machine. A pair of *equivalent machines* will be understood to be any two machines which meet Moore's criterion of isomorphism [1]. That is, if both are concealed in black boxes, there is no experiment an external observer can perform which will permit a

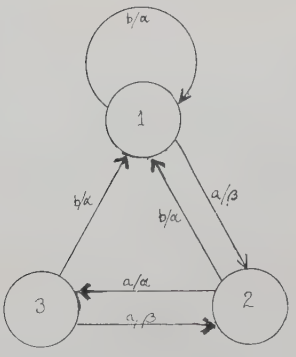


Fig. 1—Moore's machine C.

distinction to be drawn between the two machines. A *minimal* equivalent machine is one having 1) the fewest possible states, and 2) the minimum number of *gates*.

The main procedure begins with a transition diagram; any data which can be translated into a transition diagram can be subjected to the algorithm of deriving a minimal sequential machine which will conform with the data. Transition diagrams can be developed from specifications of machine responses to events or patterns of input sequences [2] or from descriptions of machines themselves [1], [6], [7]. We will find that, in general, a resulting minimal sequential machine is not unique. It is one of a set of machines of equivalent minimality, having the same number of components or *gates*. The operation of a gate is defined as a combinational function; *i.e.*, it can be represented by a Boolean form [3]–[5], [8]. Each member of the set of equivalent minimal machines will have a different collection of gates. Whether all or any of the equivalent minimal machines have gates which can be realized with desired conciseness under contemporary techniques is another question which will be touched upon.

The class of machines to be investigated is that defined by Mealy's model of a sequential circuit [6], the essential characteristics of which are the following. Let the finite number of (binary) inputs and outputs be $x_1, x_2, \cdots, x_g$ and $y_1, y_2, \cdots, y_h$ respectively; let s' be the *next state* of a machine whose *present state* is s. If X and Y are the present input and output combinations respectively, then $Y=\phi(X, s)$ and $s'=\chi(X, s)$, where ϕ and χ belong to a set of functions Σ. It is our task to determine the precise nature of Σ.

Reduction of Transition Diagrams

The first step in the procedure is to reduce the transition diagram to its simplest equivalent; this consists of merging the states of the machine into the fewest possible new states. The process to be used here is based on the matrix method of Aufenkamp and Hohn [7], which applies on the rows and columns of a matrix and may therefore be considered two-dimensional, but it has been used with the same results in operations on a one-dimensional table which contains the same information as the transition diagram. First it is necessary to establish a definition.

Definition 1: Two states of a machine are *equivalent* if and only if their outputs are the same and their next states are either identical or equivalent in turn for any input combination.

This definition implies suspended judgment in the case of certain pairs of states whose outputs are identical but where it is not immediately clear whether the next states will be equivalent. It may be necessary to examine one or more chains of *referent* pairs of states to see if they finally lead to identical pairs. If they all do, the equivalence of the pair in question is established. Some chains may close upon themselves, which is a valid demonstration of equivalence. Any set of equivalent states may be replaced by a single representative state.

As an introductory example, let us consider Moore's Machine C [1], modified in Fig. 1 to conform with Mealy's definition by assuming that the output associated with each state begins with the transition of the machine into that state. Instead of q_1, q_2, q_3, states are designated only by the subscripts 1, 2, and 3. The notation b/α means that for input b output α, occurs together with the indicated transition to the next state. The following table (which is similar to those used by Mealy [6]) contains the same data as the diagram.

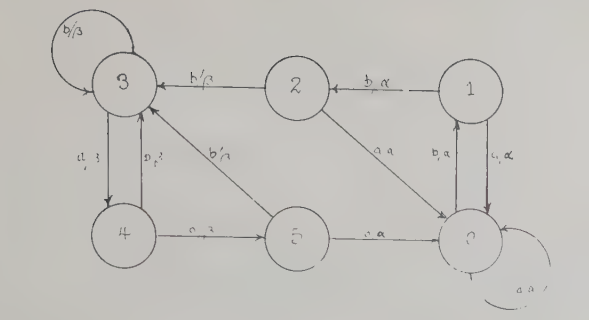
Present State	Output		Next State	
	a	b	a	b Input
1	β	α	2	1
2	α	α	3	1
3	β	α	2	1

The columns under Output give the output response associated with specific inputs of a or b. The relationship is the same in the columns under Next State. This information could be combined, but it is separated so that the two reduction steps may be identified and executed independently. It may be seen by inspection of the above table that states 1 and 3 are identical, and one of them can therefore be eliminated. This agrees with Moore's analysis.

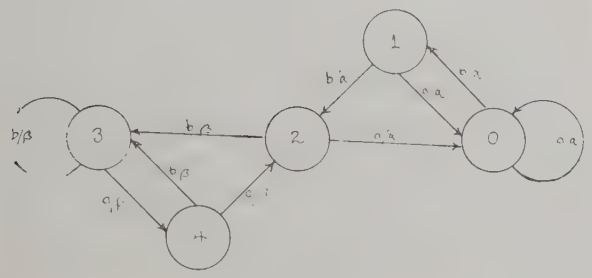
Fig. 2(a) is a transition diagram for a machine S_3 whose behavior is as follows. For any input sequence of at least three consecutive a's, the output of the machine goes to α and remains there until a sequence of at least three consecutive b's occurs, whereupon the output goes to β, etc. The *transition table* for S_3 follows.

State	Output		Next State	
	a	b	a	b
0	α	α	0	1
1	α	α	0	2
2	α	β	0	3
3	β	β	4	3
4	β	β	5	3
5	α	β	0	3

The rows of the table are now rearranged so that identical outputs are adjacent, and the table is partitioned as shown.



(a)



(b)

Fig. 2—Machine S_3.

State	Output		Next State	
	a	*b*	*a*	*b*
0	α	α	0	1
1	α	α	0	2
2	α	β	0	3
5	α	β	0	3
3	β	β	4	3
4	β	β	5	3

It is clear that states 2 and 5 are identical in their behavior, and the latter has been eliminated by superimposing their circles in the transition diagram of Fig. 2(b). All other pairs of states, within partitions, must be tested for equivalence. This is done in a mechanical way under the following rule: locate successively the pairs of next states corresponding to each pair of states within a partition. If any such pair does not, in turn, fall within a partition, nonequivalence is proved for that chain. Otherwise, if this occurs, equivalence remains indeterminate until confirmed by identical pairs or closure of the chain upon itself. For example, in order that states 3 and 4, above, be equivalent, states 4 and 5 (in the Next State columns) must be equivalent. But 4 and 5 (in the State column) are in different partitions, which proves the nonequivalence of 3 and 4, under Definition 1.

For larger machines, it is convenient to write only that portion of the table which is being treated in each step. This is done in Fig. 3(a) and (b), which are taken from the matrix on page 283 of Aufenkamp and Hohn [7]. Only the output data are transcribed in Fig. 3(a); this determines the partitioning of Fig. 3(b) and need not be recopied. Chains of state-pairs are sufficiently complicated here to justify the use of another tabulation, Fig. 3(c), in which the boxes indicate the possible pairs of states within each of the partitions. We go through Fig. 3(b), placing an *X* for each pair found to be *nonequivalent*, marking the most obvious cases first. The remaining empty boxes are necessarily those of equivalent state-pairs. In Fig. 3(c), the sets of equivalent states are 1–9–12, 3–4, 7–10, 2–8, and 6–13, which agrees with the findings of the paper cited.

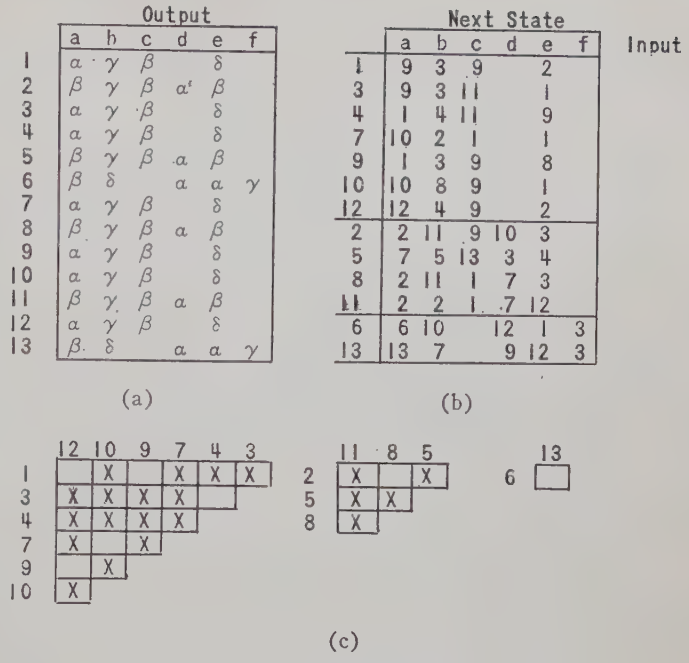


Fig. 3—Machine of Aufenkamp and Hohn.

Combination Tables

The information contained in the transition table for a diagram like that of Fig. 4, Machine *M*, can be rearranged into a *combination table*.

State		Input	Next State		Output
p	*q*	*x*	*p'*	*q'*	*y*
0	0	0	0	1	0
0	0	1	1	0	1
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	0	0	1
1	0	1	0	1	0

The columns labeled *p* and *q* form the binary equivalents of the decimal *state* numbers. Columns *p'* and *q'* are the *next state*, *x* is the input, and *y* is the output. The first row of the table then shows that state 00 is sent by input 0 into state 01, with an output of 0, and similarly for the other rows. In Fig. 5, symbol $-/0$ means that any input sends state 01 into 10, with output 0, which data also appear in the table above.

If we regard the entries of a column such as *p'* as the outputs of a gate or combinational net, then we can label the gate *P*, and say that *P* has an output of "1" when and only when there is a "1" in column *p'*. The same observations apply to *q'*, to *y*, etc.; there will be one gate in the machine for each column in the right half of

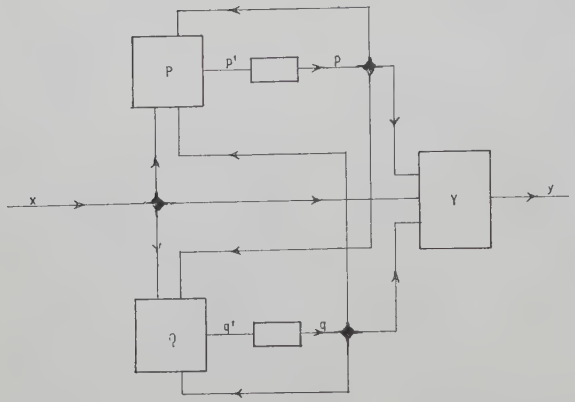


Fig. 4—Machine M, general minimal network.

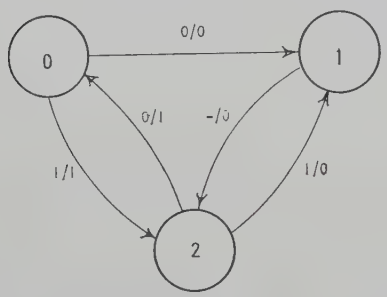


Fig. 5—Machine G.

the table. P has output 1 when any of the three combinations in the left half of the table

$$\begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{matrix}$$

occurs; otherwise, it has output 0. This matrix will be recognized as one way of representing a combinational function, the subject of which has been analyzed in some detail elsewhere [4], [8], [9]. For Q and Y, the functions are

$$\begin{matrix} 0 & 0 & 0 \\ 1 & 0 & 1, \end{matrix} \qquad \begin{matrix} 0 & 0 & 1 \\ 1 & 0 & 0. \end{matrix}$$

We can now describe a Machine M, as in Fig. 4, whose gates carry out the functions P, Q, and Y, and which has two one-pulse delay lines shown by the small rectangles. If Machine M is enclosed in a black box, its behavior will be indistinguishable from that of Machine G, whatever its original structure happened to be.

However, in Fig. 5 the assignment of numbers to the three states was arbitrary. It could have been 1, 2, 3 or 0, 3, 1 or, in fact, any ordered triple of the set $\{0, 1, 2, 3\}$, of which there are 24. Elements of this set are restricted to binary numbers which can be written in not more than two columns, by our second criterion of minimality (the least possible number of gates). Does permuting the elements of these triples affect the nature of the functions derived as we have done just above? Most certainly it does. The permutation (13) produces

$$P = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{matrix} \qquad Q = \begin{matrix} 0 & 0 & 0 \\ 1 & 0 & 1 \end{matrix} \qquad Y = \begin{matrix} 0 & 0 & 1 \\ 1 & 0 & 0, \end{matrix}$$

in which Q and Y are unaltered, but P is distinctly a different function. Such permutations affect only columns p, q, p', q' and not x or y. It might appear essential to exhaust the 24 permutations to discover the aggregate of the available alternative sets of gates, but this is not necessary. Moreover, even going through that complete process would *not* develop all possible sets of gates which comprise machines of potentially equivalent minimality (having not more than three gates). Of these sets, there may be some requiring less than three gates, and it will be shown how the existence or nonexistence of these can be determined. Some results of [4] which will be useful in the continuation will first be summarized.

Function Types

It is well known that any Boolean form can be represented as a subset of the rows of a matrix such as Fig. 6. For instance, the matrix

$$\begin{matrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{matrix}$$

corresponds to the Boolean form $a\bar{b}c \vee ab\bar{c}$. These submatrices occur in sets, the elements of which are *symmetry variants* of each other; the symmetries being those of complementation or permutation of the variables a, b, c, $\cdots$. Any complementation or permutation of the columns of the matrix is isomorphic with the same operation on the variables of the algebraic form. As McCluskey [8] and Slepian [10] have pointed out, such variations do not change the transmission of the gate, since they correspond to inversion or rearrangement of its input connections. Forms which are symmetry variants of each other are said to be of the same *type;* the definition of type is extended here to include inversion of the gate output, which corresponds to taking the over-all complement of the algebraic form. On a matrix like Fig. 6, the corresponding operation is to take all rows *not* in the given submatrix. When a submatrix has exactly half of the possible rows (*e.g.*, four rows of Fig. 6) it is a symmetry variant of its over-all complement if there are fewer than four columns, but with four or more this is not generally true. However, it will always be of the same type, by definition.

It is possible to identify all forms of a given type by reference to a *unique symmetry variant.* If many identifications are to be made, it is convenient to set up a list like that of Table I to eliminate repeated use of the procedure of [4] to identify function types. To employ the table, choose any set of column complementations which yields a matrix row of all zeros. Then read the

a	b	c
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Fig. 6.

p	q	x	p'	q'	y
0	0	0	0	1	0
0	0	1	1	0	1
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	—	—	—
1	1	1	—	—	—

Fig. 7—Full combination table for machine G.

TABLE I

FUNCTION TYPES OF THREE VARIABLES

1-row		023	*e*	0124	*m*	0245	*k*
(all)	*d*	24	*e*	125	*k*	246	*a*
2-row		25	*l*	126	*k*	247	*g*
01	*b*	26	*e*	127	*g*	256	*g*
2	*b*	27	*l*	134	*k*	257	*c*
3	*f*	34	*l*	135	*m*	267	*k*
4	*b*	35	*j*	136	*g*	345	*g*
5	*f*	36	*j*	137	*k*	346	*g*
6	*f*	37	*l*	145	*a*	347	*k*
7	*h*	45	*e*	146	*k*	356	*i*
		46	*e*	147	*g*	357	*g*
3-row		47	*l*	156	*g*	367	*g*
012	*e*	56	*j*	157	*k*	456	*m*
13	*e*	57	*l*	167	*c*	457	*k*
14	*e*	67	*l*	234	*k*	567	*g*
15	*e*			235	*g*		
16	*l*	4-row		236	*m*		
17	*l*	0123	*a*	237	*k*		

decimal equivalents of the rows in numerical order, and locate this sequence in the table. The bold-face letter represents the *associated function* [3]. The machine designer will require another table giving the algebraic equivalents of these functions, perhaps like the tables of [9] or [4] or those of Table III, to follow. Otherwise, he can take the normal disjunctive forms directly from the matrices and work with these, possibly using such recent techniques as those of Ghazala [12]. In any event, he will have a table of functions suited to his purpose, and it needs to contain only one formula for each symmetry type.

Let us rewrite the combination table for Machine G, placing the full matrix of Fig. 6 on the left. This is represented in Fig. 7. The short dashes in Fig. 7 indicate that those values may be assigned or omitted in any arbitrary manner. The states which are below the partition in such a table can never occur in the machine; hence we may suppose any response at all, and it need not be consistent among P, Q, and Y. This latitude can greatly simplify the gate function. If the *essential rows* are those above the partition, and for a particular machine K these happen to be

p	q	x
0	0	1
0	1	1
1	0	1

in the rows where column y is 1, then function Y would be $x(\bar{p} \vee \bar{q})$. But if there is also a *nonessential row* 1 1 1 below the partition, it can be added to the matrix; whereupon Y becomes simply x, and machine K may be replaced by a "gate" which is a permanent connection between input and output. Data like those of Fig. 7 may be extended to accommodate multiple inputs $x_1, x_2, \cdots, x_g$ or multiple outputs $y_1, y_2, \cdots, y_h$. Each additional x_i adds a column and doubles the length of the table, but each additional y_j only adds one column.

SETS OF EQUIVALENT MACHINES

In Fig. 7, the functions are

$$P = \begin{array}{ccc} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ \hline 1 & 1 & 0 \\ 1 & 1 & 1 \end{array} \qquad Q = \begin{array}{ccc} 0 & 0 & 0 \\ 1 & 0 & 1 \\ \hline 1 & 1 & 0 \\ 1 & 1 & 1 \end{array} \qquad Y = \begin{array}{ccc} 0 & 0 & 1 \\ 1 & 0 & 0 \\ \hline 1 & 1 & 0 \\ 1 & 1 & 1. \end{array}$$

Then by referring to Table I and using 1) only essential rows, 2) the upper nonessential row, 3) the lower, and 4) both nonessential rows, we can determine that the function types are

$$P = \mathbf{e, k, m, e} \qquad Q = \mathbf{f, j, l, g} \qquad Y = \mathbf{f, l, j, g.}$$

Although Q and Y are of the same type, they are not identical gates, since they differ in the complementation of the x input. Now, using decimal equivalents, let (02) mean that states 00 and 10 are to be permuted. In Fig. 7, this would require writing "00" wherever "10" occurs in the left pair of columns and conversely. The same applies to (01), (13), etc. If (I) is the identity permutation, then making the (01) and (02) permutations independently on Fig. 7 produces the results found in Table II. In this table, each sequence of four letters lists the function types derived from taking the nonessential rows in the order previously given. Repetitions are shown because they represent alternate symmetries of the same function type; this information might be helpful in completing the design of a circuit or machine (one or more input or output inversions might be eliminated). The question now to be resolved is how many more of the 24 permutations of states must be made to complete Table II. An abbreviated proof will be given that the answer is *none;* every further permutation

TABLE II
FUNCTION TYPES FOR MACHINE G

	P	Q	Y
(I)	*e, k, m, e*	*f, j, l, g*	*f, l, j, g*
(01)	*e, g, k, l*	*d, b, f, e*	*h, l, l, k*
(02)	*d, f, h, l*	*f, l, j, g*	*f, j, l, g*

will yield some rearrangement of a row of Table II which is *imprimitive*. That is, the letters of a quadruple may be permuted among themselves, or quadruples may be interchanged laterally, but there will be no cross-mingling or appearance of new letters.

Proof: Let the initial ordering of the four states be 0123. The set of all permutations of these objects among themselves forms the *symmetric group* of order 4, which Ledermann [12] calls P_4. This group can be generated by the transpositions (01), (02), (03) and products thereof ([12], Theorem 7, page 76). The identity permutation, (01), and (02) have been accounted for in Table II.

Consider the effect of symmetry variations upon any (full) table like that of Fig. 7. It may be readily verified that complementation of any variable p, q, r, $\cdots$, or permutation among them does not change the functions P, Q, R, $\cdots$, beyond requiring the over-all complement or permuting P, Q, R, $\cdots$, which does not affect the function type, by definition.

On 0123, the symmetry variations—complementing p, complementing q, or the permutation (pq)—are 2301, 1032, and 0213, respectively. Let these elements of P_4 be designated κ, λ, and μ; and let (01), (02), and (03) be α, β, and γ, respectively. Since subsequent symmetry variations do not alter function types, those elements of P_4 which are produced by post-multiplication of I, α, or β by κ, λ, or μ do not create rows whose composition is distinct from those of Table II. (Pre-multiplication is forbidden because it can have a transforming effect: $\mu\alpha\mu=\beta$.)

Permutation (03) is made unnecessary by $\kappa\lambda\mu=\gamma$. Furthermore, the products of α and β are unnecessary by the relations $\beta\mu=\alpha\beta$ and $\alpha\mu=\beta\alpha$. In fact, the 24 elements of P_4 can be shown to have the *faithful representation*,

I	γ	$\beta\lambda$	$\alpha\kappa\lambda$
α	$\alpha\kappa$	$\beta\kappa\lambda$	$\beta\kappa\lambda$
β	$\alpha\lambda$	$\beta\gamma$	$\alpha\kappa\mu$
κ	$\alpha\mu$	$\kappa\lambda$	$\beta\kappa\mu$
λ	$\alpha\gamma$	$\kappa\mu$	$\alpha\lambda\mu$
μ	$\beta\kappa$	$\mu\kappa$	$\beta\lambda\mu$,

which demonstrates that post-multiplication of I, α, or β by combinations of κ, λ, and μ are sufficient to generate all elements of the group. This completes the proof.

MACHINE STRUCTURES

Having obtained data in the form of Table II, the remaining task for the designer is to select the exact set of gates to be assembled into a machine. The selection will be affected by special criteria, of which one of the most significant will be the availability of gates. If he can choose a set of gates, each of which operates directly, within one pulse time, his timing problems for the whole machine are simplified, and hazard or race conditions [13] are likely to be minimal. If he has only relays or the usual set of AND, OR, and NOT gates at his disposal, it will be advantageous to consider not only which sets of functions (Table II) are the simplest, but also which rows of the function matrices are identical and may therefore be shared. An extensive study of this particular problem could be made.

The selection task is simplified as the stockpile of available gate types is enlarged. If the designer has components on hand which will perform each of the polyadic relations of three variables, then he can assemble any machine having not more than four states in a manner similar to that of machine M (Fig. 4), which has at most three functional gates. Efforts now in progress are contributing toward such a stockpile [5], [14], [15]. At least two contracts[1] now in progress have, as partial objectives, the construction of 3-variable gates which have not yet been realized, in composite blocks of materials. A superb example of such a block is the Rutz transistor [16], [17], whose two outputs are triadic functions (types ***i*** and ***m*** in Table III). The polyadic functions of [4] are given in more familiar notation here in Table III.

TABLE III
EXAMPLES OF FUNCTIONS OF THREE VARIABLES

Type	Algebraic Form	Type	Algebraic Form
a	p	***h***	$pqr\vee\bar{p}\bar{q}\bar{r}$
b	pq	***i***	$p\equiv(q\equiv r)$
c	$p\equiv q$	***j***	$pq\bar{r}\vee p\bar{q}r\vee\bar{p}qr$
d	pqr	***k***	$\bar{p}q\bar{r}\vee p(q\vee r)$
e	$p(q\vee r)$	***l***	$pqr\vee\bar{q}\bar{r}$
f	$p(q\equiv r)$	***m***	$pq\vee pr\vee qr$
g	$p\equiv(qr)$	***n***	0 (no transmission)

The forms in the table are merely representative examples of the various types of functions. They should not be confused with unique symmetry variants [4].

Returning to the case of machine G, we note that Table II does not offer any gate types which have less than three input variables (this is the next step in the design procedure). The designer now knows he cannot obtain a structure simpler than that of Fig. 5, with three gates having three inputs each. Whatever is in his stockpile largely determines his next action. He may want as few input and output inversions as possible, or he may not care. Special considerations arise: a non-equivalence gate (type **c** with one variable complemented) may be easier to build than an equivalence

[1] Contracts AF 33(616)6416, with the Ohio State University Research Foundation, Columbus, Ohio; and AF 33(616)6355, with the Burroughs Corp., Paoli, Pa.

gate; in a magnetic core, one symmetry variant of type ***m*** is as easy as another. In general, the designer is likely to be more concerned over complements than permutations of variables. For the sake of an example, let us assume that he wants to use only the shorter forms of Table III denying himself types ***h, j, k, l,*** and ***m,*** and that he desires to employ the smallest number of dyadic gates as well as the least number of inversions (which may be conflicting objectives). Then the aggregate of forms from which he can choose a set is as shown in Table IV. Any triple from the upper or lower half of the table may be used. Any variable may be complemented in this table, provided that all of its occurrences are so treated. However, if p or q is complemented, p' and q' are similarly affected so that either the output of P must be inverted (in the case of p') or the over-all complement must be applied to the forms under P in the table. With this in mind, we can obtain the following explicit set of gate descriptions:

$$p' = p \vee \bar{q} \vee x, \quad q' = \bar{q} \vee (p \equiv x), \quad y = q(p \equiv x).$$

TABLE IV
SETS OF FORMS FOR MACHINE G

P	Q	Y
$\bar{p}(q \vee \bar{x})$ $q \vee \bar{p}x$	$\bar{q}(p \equiv x)$ $\bar{p} \equiv \bar{q}\bar{x}$	$\bar{q}(p \equiv \bar{x})$ $\bar{p} \equiv \bar{q}x$
$\bar{p}\bar{q}\bar{x}$ $\bar{x}(p \equiv q)$	$\bar{q}(p \equiv \bar{x})$ $\bar{p} \equiv \bar{q}x$	$\bar{q}(p \equiv x)$ $\bar{p} \equiv \bar{q}\bar{x}.$

There are only two inversions here, one on an input and one on an output, and the machine can be built with four functional components (if we permit a three-input OR gate), a fairly compact result in view of the restrictive conditions which were assumed.

Large Machines

In the left half of a combination table, as in Fig. 7, there must be a column for each of the g machine inputs, and k additional columns are necessary in general for a machine having a number of states s such that $2^{k-1} < s \leq 2^k$. If c is the total number of columns on the left of the table, then

$$c \leq g + \log_2 s,$$

and the number of rows in a full table is 2^c. If h is the number of machine outputs, then the number of functional gates f required is

$$f \leq h + k.$$

For a machine in which $k = 3$, the permutation group P_8 can be generated by the transpositions (01), (02), (03), (04), (05), (06), and (07). These transpositions are all *odd* permutations, whereas the symmetry variations here are all *even* permutations; therefore, it can be stated immediately that no product of the latter set can generate any element of the former. The order of P_8 is 40,320; and if S is the subgroup of all of the symmetry variants, then S can be shown to be of order 48, which means that the index of S in P_8 is 840. P_8 can be decomposed into 840 left-hand cosets relative to S, and there exists a set of 840 products of transpositions which is sufficient to determine all distinct functions of the system. (In P_4, S was of index 3.) Similar analysis may be employed for larger systems. It is clear that the labor involved in the full analysis of a large machine will generally require the assistance of a high-speed computing facility.

Conclusion

Through the use of a simple example, a procedure has been outlined for deriving from any given transition diagram a set of machines of equivalent minimality. On the basis of functional types, there exist sets of machines having not more than $h+k$ gates for every diagram having not more than 2^k states where there are h outputs. If there are machines in these sets which have fewer than $h+k$ gates, the procedure will disclose such possibilities, since all available types of functions are surveyed.

Bibliography

[1] E. F. Moore, "Gedanken-experiments on sequential machines," in "Automata Studies," Princeton University Press, Princeton, N. J., pp. 129–153; 1956.

[2] I. M. Copi, C. C. Elgot and J. B. Wright, "Realization of events by logical nets," *J. Assoc. Comp. Mach.*, vol. 5, pp. 181–196; April, 1958.

[3] A. Church, "Introduction to Mathematical Logic," Princeton University Press, Princeton, N. J.; 1956.

[4] D. B. Netherwood, "Logic matrices and the truth function problem," *J. Assoc. Comp. Mach.*, vol. 6, pp. 405–414; July, 1959.

[5] U. F. Gianola and T. H. Crowley, "The Laddic—a magnetic device for performing logic," *Bell Sys. Tech. J.*, vol. 38, pp. 45–72; January, 1959.

[6] G. H. Mealy, "A Method for Synthesizing Sequential Circuits," Bell Telephone System Monograph 2458; also, *Bell Sys. Tech. J.*, vol. 34, pp. 1045–1079; September, 1955.

[7] D. D. Aufenkamp and F. E. Hohn, "Analysis of sequential machines," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 276–285; December, 1957.

[8] E. J. McCluskey, Jr., "Minimization of Boolean functions," *Bell Sys. Tech. J.*, vol. 35, pp. 1417–1444; November, 1956.

[9] H. H. Aiken and staff of the Computation Lab., Harvard University, "Synthesis of Electronic Computing and Control Circuits," Harvard University Press, Cambridge, Mass.; 1951.

[10] D. Slepian, "On the number of symmetry types of Boolean functions of N variables," *Canad. J. Math.*, vol. 5, no. 2, pp. 185–193; 1953. Also, Bell Telephone Labs., Inc., Monograph 2154; February, 1954.

[11] M. J. Ghazala, "Irredundant disjunctive and conjunctive forms of a Boolean function," *IBM J. Res. & Dev.*, vol. 1, pp. 171–176: April, 1957.

[12] W. Ledermann, "Introduction to the Theory of Finite Groups," Oliver and Boyd, Ltd., Edinburgh, Scotland, and Interscience Publishers, Inc., New York, N. Y.; 1957.

[13] D. A. Huffman, "The design and use of hazard-free switching networks," *J. Assoc. Comp. Mach.*, vol. 4, pp. 47–62; January, 1957.

[14] H. W. Abbott and J. J. Suran, "Multihole ferrite core configurations and applications," PROC. IRE, vol. 45, pp. 1081–1093; August, 1957.

[15] H. D. Crane, "A high-speed logic system using magnetic elements and connecting wire only," PROC. IRE, vol. 47, pp. 63–73; January, 1959.

[16] R. F. Rutz, "Two-collector transistor for binary full addition," *IBM J. Res. & Dev.*, vol. 1, pp. 212–222; July, 1957.

[17] B. Dunham, "The multipurpose bias device. Part I. The commutator transistor," *IBM J. Res. & Dev.*, vol. 1, pp. 116–129; April, 1957.

A Technique for the Reduction of a Given Machine to a Minimal-State Machine*

SEYMOUR GINSBURG†

Summary—**A technique is presented for reducing an arbitrary machine S as much as possible to a machine T which can do everything (from the input-output point of view) that S can do. Since the technique is always applicable, it is more powerful (although more cumbersome) than the well-known merging technique. Several examples are given.**

Introduction

THE following problem has been introduced [2]:

Problem Q

Suppose that a machine S (not necessarily completely defined for each state q and each input I) is given. Find a machine T satisfying the following two properties:

1) $S \leqq T$. By this it is meant that for each state p of S there is a state $f(p)$ in T so that $p \leqq f(p)$. That is, each sequence of inputs which can be applied to machine S starting at state p can also be applied to machine T starting at state $f(p)$, such that the two sequences of resulting outputs so obtained are identical.

2) The number of states in T is not greater than the number of states in any other machine Y for which $S \leqq Y$ holds.

In less formal language, Problem Q consists in trying to reduce a given machine S as much as possible to a machine T which can do everything that S can do. This problem has been attacked via the merging technique [1], [3]–[5]. By merging in all possible ways, a set of machines is obtained. From this set, a machine having the fewest number of states is selected. However, the machine so selected is *not* necessarily a solution to Problem Q. As noted in [2], there is at present no "practical" general method of solving Problem Q for an arbitrary machine S. All practical results obtained in [2] pertain to the lesser problem of general reduction (of superfluous states); *i.e.*, in going from S to Y, where $S \leqq Y$. In this paper a technique is presented which, in theory at least, always leads to a solution to Problem Q.

This technique involves the following steps:

1) A particular lower bound, say h, on the number of states of any machine Y for which $S \leqq Y$ holds is determined.
2) Each machine Y with h states is tested to see if $S \leqq Y$ holds.
3) If no h-state machine Y satisfying $S \leqq Y$ exists, then step 2) is repeated for $h+1$, $h+2$, etc., states.
4) The process is terminated when some machine Y with $h+g$ states, g a non-negative integer, is found such that $S \leqq Y$.

Clearly, the machine found in step 4) is a solution to Problem Q.

The execution of 1) is automatic and is easily programmed on a computer. The execution of steps 2) and 3) is not automatic; it depends on the enumeration and the selection of numerous alternative paths of procedure. The more intelligent the selections, the less the resulting computation. Furthermore, the selection of a different sequence of alternatives may lead to a different solution. In order to render steps 2) and 3) automatic, an order of preference in the determination of the alternatives (that is, a definite set of rules which determines the alternatives) must be made. As yet, this has not been done; thus the method presented is more of an approach than an algorithm or a mechanical procedure.

I. Preliminaries

By a (deterministic) machine S is meant a finite number of inputs $I_1, \cdots, I_m$; a finite number of (internal) states $q_1, \cdots, q_n$; a finite number of outputs $E_1, \cdots, E_r$; and two functions δ_S and λ_S, written δ and λ for short when S is understood. δ is called the *next state* function and λ, the *output* function. For some (possibly all) pairs (q, I), q a state and I an input, δ takes (q, I) into a state $\delta(q, I)$; and for some (possibly all) pairs of (q, I), λ takes (q, I) into an output $\lambda(q, I)$. Given the inputs, the outputs, and the states, a machine is thus determined when δ and λ are specified for certain (q, I). A machine is usually given in matrix form as shown in Fig. 1. [Some of the rectangles have a δ or a λ entry (or both).] A matrix as given in Fig. 1 is called a δ, λ *matrix*.

Let $I_1, \cdots, I_k$ be a sequence of inputs to a machine. $\lambda(q_1, I_1 \cdots I_k)$ is said to exist if $\lambda(q_1, I_1), \cdots, \lambda(q_k, I_k)$, and each $q_{i+1} = \delta(q_i, I_i)$ for $i < k$ exist; and $\lambda(q_1, I_1 \cdots I_k)$, when it exists, is defined to be $\lambda(q_1, I_1)\lambda(q_2, I_2) \cdots \lambda(q_k, I_k)$. $\delta(q_1, I_1 \cdots I_k)$ is said to exist if each state $q_{i+1} = \delta(q_i, I_i)$ for $i \leqq k$ exists; and when $\delta(q_1, I_1 \cdots I_k)$ exists, it is defined to be the state q_{k+1}. Thus $\lambda(q_1, I_1 \cdots I_k)$ is the output obtained when the

* Manuscript received by the PGEC, January 12, 1959; revised manuscript received, March 23, 1959.

† Hughes Research Labs., P.O. Box 90427, Los Angeles, Calif. Formerly with National Cash Register Co., Hawthorne, Calif.

	q_1	$\cdots$	q_j	$\cdots$	q_n
I_1	$\delta(q_1, I_1)$				
$\vdots$					
I_i			$\delta(q_j, I_i)\lambda(q_j, I_i)$		
$\vdots$					
I_m					$\lambda(q_n, I_m)$

Fig. 1—Typical δ, λ matrix.

sequence of inputs $I_1, \cdots, I_k$ is applied to the machine initially in state q_1; and $\delta(q_1, I_1 \cdots I_k)$ is the state to which the machine goes from state q_1 upon insertion of the sequence of inputs $I_1, \cdots, I_k$.

Observe that

$$\lambda(q_1, I_1 \cdots I_k) = \lambda(q_1, I_1 \cdots I_j)\lambda(\delta(q_1, I_1 \cdots I_j), I_{j+1} \cdots I_k)$$

whenever either side exists, and

$$\delta(q_1, I_1 \cdots I_k) = \delta(\delta(q_1, I_1 \cdots I_j), I_{j+1} \cdots I_k)$$

whenever either side exists.

A sequence of inputs $I_1, \cdots, I_k$ is said to be *applicable* to a state q_1 if $\lambda(q_1, I_1 \cdots I_k)$ exists.

Notation

Let S and T be two machines. For states p in S and q in T, by $p \leqq q$ is meant that each sequence $I_1, \cdots, I_k$ of inputs which is applicable to p is applicable to q and, in addition, $\lambda_S(p, I_1 \cdots I_k) = \lambda_T(q, I_1 \cdots I_k)$. Given two machines S and T, by $S \leqq T$ is meant that to each state p in S there exists a state $f(p)$ in T such that $p \leqq f(p)$.

Roughly speaking, for states p in S and q in T, $p \leqq q$ means that everything that machine S can do starting at p, machine T can do starting at q. $S \leqq T$ means that machine T can do everything that machine S can do.

Given two machines S and T, a "practical" test for determining whether or not $S \leqq T$ is given by Theorem 1 below. Since this result is not used in the sequel, its proof, which is similar to that of Theorems 3 and 4 and Lemmas 2 and 3 of [2], is omitted.

Theorem 1

Let S and T be two machines with n and m states, respectively. Let P_0 be the set of all ordered pairs (p, q) of states p in S and q in T. Let P_1 be the set of all pairs (p, q) in P_0 such that whenever $\lambda_S(p, I)$ exists, $\lambda_T(q, I)$ both exists and is $\lambda_S(p, I)$. For each integer $k>1$ let P_k be the set of all pairs (p, q) in P_{k-1} with the property that if $\delta_S(p, I)$ exists and if $\lambda_S[\delta_S(p, I), I^*]$ exists for some input I^*, then $\delta_T(q, I)$ exists and $[\delta_S(p, I), \delta_T(q, I)]$ is in P_{k-1}. Then

a) $S \leqq T$ if and only if: $P_{mn-n} = P_{mn-n+1}$; for each state p in S there exists a state q in T so that (p, q) is in P_{mn-n}.

b) for any integer $j \leqq mn-n$, if $P_j = P_{j+1}$, then $P_j = P_{j+i} = P_{mn-n} = P_{mn-n+1}$ for every positive integer i.

Remark

Given two positive integers m and n, there exist two machines S and T, with n and m states respectively, such that $S \leqq T$ and $P_i \neq P_{mn-n}$ for each positive integer $i<mn-n$. Thus the numbers $mn-n$ and $mn-n+1$ in a), in general, cannot be lowered. To see this, let n and m be any two positive integers. Clearly, it may be assumed that $m>1$. S is to be a machine with n states $p_1, \cdots, p_n$; n inputs $I_1, \cdots, I_n$; and one output E_1. Let $\lambda_S(p_i, I_i) = E_1$ for $i \leqq n$, let $\delta_S(p_i, I_i) = p_{i+1}$ for $i<n$, and let $\delta_S(p_n, I_n) = p_1$. T is to be a machine with m states $q_1, \cdots, q_m$; n inputs $I_1, \cdots, I_n$; and two outputs E_1 and E_2. Let $\delta_T(q_1, I_j) = q_1$ for $j \leqq n$, let $\delta_T(q_i, I_j) = q_i$ for $1<i \leqq m$ and $j<n$, let $\delta_T(q_i, I_n) = q_{i+1}$ for $1<i<n-1$, and let $\delta_T(q_m, I_n) = q_2$. Let $\lambda_T(q_i, I_j) = E_1$ for $i<m$ and $j \leqq n$, let $\lambda_T(q_m, I_j) = E_1$ for $j<n$, and let $\lambda_T(q_m, I_n) = E_2$. It is easily verified that

$$P_{mn-n} = P_{mn-n+1} = \{(p_i, q_1) \mid 1 \leqq i \leqq n\},$$

and that $P_i \neq P_{mn-n}$ for $i<mn-n$.

Problem Q may be stated as: given a machine S, find a machine T with as few states as possible such that $S \leqq T$. A general discussion of this problem is given in [2]. At present, there is no "practical" procedure for finding a solution to Problem Q when the given machine S is perfectly arbitrary. The object of this paper is to introduce a technique which, in theory at least, always leads to a solution of Problem Q. In outline form this technique consists of steps 1), 2), 3), and 4) of the introduction. As is usual for general methods, the details of this technique are best explained by applying it to solve Problem Q for some specific machines S.

	p_1	p_2	p_3	p_4	p_5	p_6
a	p_5 3	p_6 3	p_4 3	p_2 4	p_1 4	p_3 4
b		p_6 1	p_4 2			
c	p_4 3			p_2 1		

Fig. 2—Machines S of Example 1.

As a first example, let S be the machine given in Fig. 2. The details involved in executing steps 1)–3) of the introduction, especially as applied to the first example, are given in the next two sections.

II. The Execution of Step 1)

The first step in the technique is to derive a lower bound, say h, on the number of states in any machine Y for which $S \leqq Y$ holds. Obviously, $h=1$ always serves as one such value. In order to keep the amount of computation associated with the execution of steps 2)–3) to a minimum, it is desirable to select as large a value for h as is possible. The method used in this paper to determine a value for h depends on material now to be presented.

Definition

Given two machines S and T, the states p in S and q in T are said to be "compatible" if $\lambda_S(p, I_1 \cdots I_k) = \lambda_T(q, I_1 \cdots I_k)$ for every sequence of inputs $I_1, \cdots, I_k$ which is applicable to both p and q [1]. If p and q are not compatible, written $p \phi q$, then p and q are said to be "incompatible."

It is readily seen that if x, y, p, and q are states such that $x \leqq p$, $y \leqq q$, and p and q are compatible, then x and y are compatible. Thus if x, y, p, and q are states such that $x \phi y$, $x \leqq p$, and $y \leqq q$, then $x \phi q$, $y \phi p$, and $p \phi q$. This implies that if $\{x_i \mid 1 \leqq i \leqq e\}$ is a set of pairwise incompatible states of S, and if $S \leqq Y$, then Y has at least e states. Consequently, if *D is a numerically maximal set of incompatible states of S, that is, if D is a set of pairwise incompatible states of S with the property that the number of elements $n(D)$ in D is as large or larger than the number of states in any other set of pairwise incompatible states of S; then $n(D)$ may serve as a value for h.* The value for h to be used in step 1) is the number of elements in a numerically maximal set of incompatible states of S.

In order to find a numerically maximal set of incompatible states of S, all pairs of incompatible states are first determined. The incompatible pairs of states are found after the compatible pairs of states are found. The compatible pairs of states are found by means of the following result noted in [2].

Theorem 2

Let S be a given machine with n states. Denote by P_1 the set consisting of all unordered pairs of states (x, y) of S such that $\lambda(x, I) = \lambda(y, I)$ for each input applicable to both x and y. In general, let P_{k+1} be the set which consists of all pairs of states (x, y) in P_k such that $[\delta(x, I), \delta(y, I)]$ is in P_k for each input I for which both $\delta(x, I)$ and $\delta(y, I)$ exist. Then x and y are compatible states if and only if (x, y) is in

$$\frac{P_{n(n-1)}}{2}$$

Furthermore, if $P_k = P_{k+1}$ for some integer k, then

$$P_k = \frac{P_{n(n-1)}}{2}.$$

Applying Theorem 2 to machine S of Example 1, the pertinent sets P_i are as follows:

2.1) $P_1 = H \cup \{(p_1, p_2), (p_1, p_3), (p_4, p_5), (p_4, p_6), (p_5, p_6)\}$,

where

$$H = \{(p_j, p_j) \mid 1 \leqq j \leqq 6\},$$

2.2) $P_2 = H \cup \{(p_1, p_2), (p_1, p_3), (p_4, p_5), (p_5, p_6)\}$,

and

2.3) $P_3 = P_2$.

Thus the pairs of compatible states in machine S of Example 1 are precisely those pairs in P_2. Since the pairs of incompatible states are those pairs which are not compatible, the pairs of incompatible states are

2.4) $(p_1, p_4), (p_1, p_5), (p_1, p_6), (p_2, p_3), (p_2, p_4), (p_2, p_5), (p_2, p_6), (p_3, p_4), (p_3, p_5), (p_3, p_6), (p_4, p_6)$.

For this particular example, it is relatively easy to find a numerically maximal set of incompatible states of S by some haphazard trial and error procedure. Instead of doing that, a systematic method of obtaining a numerically maximal set of incompatible states for any machine S is now described. (The method essentially consists of examining in sequence each set of incompatible states, retaining at each instant the numerically largest of all the sets encountered.) The reader who is not interested in a formal treatment of this aspect of the problem may, after agreeing that $\{p_2, p_3, p_4, p_6\}$ is a numerically maximal set of incompatible states of S, proceed immediately to Section III.

2.5) Order the states $p_1, p_2, \cdots, p_n$ of S by the magnitude of the indexes, that is $p_a < p_b$ if and only if $a < b$. This ordering also orders the elements in each set of states of S.

2.6) At each moment, two sets of incompatible states, called the A-set or A for short and the B-set or B for short, are under scrutiny. Initially A and B each contain the one state p_1. (The A-set is a numerically maximal set of incompatible states among all sets already scrutinized except possibly subsets of the B-set. The B-set is a set of incompatible states which, if joined with enough additional states, becomes the new A-set.)

2.7) To each B-set and each state p_a in B, associate the set (possibly empty) of all those states in S after p_a (that is, with indexes $> a$) which are incompatible both with p_a and with all the states in B preceding p_a (that is, with indices $< a$). Denote this set by $C(p_a)$ although, of course, it depends on the B-set as well as the state p_a.

2.8) Let p_d be the last element in B.

a) Suppose that $C(p_d)$ is nonempty and that $C(p_d) \cup B$ is numerically larger than A. Let p_e be the first element in $C(p_d)$. Define the new B-set to be the old B-set together with p_e, and let the A-set be unchanged. (This enlarges B.) Proceed to 2.7)

b) Suppose that $C(p_d)$ is empty. Then compare B numerically with A. If B is numerically larger than A, define the new A-set to be the old B. Otherwise let A be unchanged. In either case, B is unchanged. (In other words, if B cannot be extended, then compare A and B numerically and select the larger of the two sets to be the new A.) Proceed to 2.9).

c) Suppose that a) and b) are both false. Proceed to 2.9). (Since B cannot be extended to be numerically larger than A, B and its extensions are given no further consideration.)

2.9) a) Suppose that there exists a state p_d next defined. p_d is the last element in B which satisfies the following two properties: p_f being the first state after p_d in B, $C(p_d)$ contains a state, thus a first state called p_e, such that $e>f$; the set $\{p_i | i \leqq d,\ p_i \text{ in } B\} \cup \{p_i | i \geqq e,\ p_i \text{ in } C(p_d)\}$ is numerically larger than A. Define the new B-set to be $\{p_i | i \leqq d,\ p_i \text{ in } B\} \cup \{p_e\}$. Let A be unchanged. Then proceed to 2.7). (The effect of this is to select a B-set, not as yet considered, which retains many of the beginning elements of the old B-set, and which might be extended to become the new A-set.)

b) Suppose that p_d as defined in a) does not exist. Let p_a be the first element in B. If p_a is the last state in S, then A is a numerically maximal set of incompatible states and the process terminates. If p_a is not the last state in S, let the new B-set consist of the one state p_{a+1} and let A be unchanged. Then proceed to 2.7). (This means that all sets of incompatible states containing p_a as a first element have been scrutinized; the sets whose first element are p_{a+1}, if p_{a+1} exists, are to be considered next.)

On applying the above method to machine S of Example 1, the following table is obtained.

Old A	*Old B*	*Steps*	*Comments*
$\{p_1\}$	$\{p_1\}$	2.6), 2.7), 2.8)a)	$C(p_1)=\{p_4, p_5, p_6\}$
$\{p_1\}$	$\{p_1, p_4\}$	2.7), 2.8)a)	$C(p_4)=\{p_6\}$
$\{p_1\}$	$\{p_1, p_4, p_6\}$	2.7), 2.8)b)	$C(p_6)$ is empty
$\{p_1, p_4, p_6\}$	$\{p_1, p_4, p_6\}$	2.9)b)	
$\{p_1, p_4, p_6\}$	$\{p_2\}$	2.7), 2.8)a)	$C(p_2)=\{p_3, p_4, p_5, p_6\}$
$\{p_1, p_4, p_6\}$	$\{p_2, p_3\}$	2.7), 2.8)a)	$C(p_3)=\{p_4, p_5, p_6\}$
$\{p_1, p_4, p_6\}$	$\{p_2, p_3, p_4\}$	2.7), 2.8)a)	$C(p_4)=\{p_6\}$
$\{p_1, p_4, p_6\}$	$\{p_2, p_3, p_4, p_6\}$	2.7), 2.8)b)	$C(p_6)$ is empty
$\{p_2, p_3, p_4, p_6\}$	$\{p_2, p_3, p_4, p_6\}$	2.9)b)	
$\{p_2, p_3, p_4, p_6\}$	$\{p_3\}$	2.7), 2.8)c), 2.9)b)	$C(p_3)=\{p_4, p_5, p_6\}$
$\{p_2, p_3, p_4, p_6\}$	$\{p_4\}$	2.7), 2.8)c), 2.9)b)	$C(p_4)=\{p_6\}$
$\{p_2, p_3, p_4, p_6\}$	$\{p_5\}$	2.7), 2.8)b), 2.9)b)	$C(p_5)$ is empty
$\{p_2, p_3, p_4, p_6\}$	$\{p_6\}$	2.7), 2.8)b), 2.9)b)	$C(p_6)$ is empty; process terminates.

Since $\{p_2, p_3, p_4, p_6\}$ is the set A at the termination of the procedure, $\{p_2, p_3, p_4, p_6\}$ is a numerically maximal set of incompatible states of S. Thus any machine Y, such that $S \leqq Y$ holds, has at least four states.

III. Execution of Steps 2)–3)

The next item considered is a procedure which carries out steps 2)–3) of the introduction. In other words, for a given integer d, a procedure is found which determines whether or not there exists a machine Y with d states such that $S \leqq Y$; and when there are such machines Y, the procedure exhibits one.

The method given here for executing steps 2)–3) depends, in part, on the following result.

Theorem 3

Let S and Y be two given machines. Suppose that "$*$" is a binary relation between the states of S and the states of Y[1] satisfying the following two properties.

3.1) For each state p in S there is at least one state q in Y such that $p * q$ holds.

3.2) For any two states p in S and q in Y, if $p * q$ holds, then *a*) $\lambda_S(p, I)=\lambda_Y(q, I)$ for each input I which is applicable to p; and *b*) if $\delta_S(p, I)$ and $\lambda_S(p, II_1)$ exist for some input I_1, then $\delta_Y(q, I)$ exists and $\delta_S(p, I) * \delta_Y(q, I)$ holds.

Then a) for any two states x in S and y in Y, $x \leqq y$ holds whenever $x * y$ holds; and b) $S \leqq T$.

Proof:

Let $*$ be a star relation[2] between S and Y. Clearly, 3.1) and a) imply b). Hence it suffices to show that a) is true.

To this end, let x in S and y in Y be two states such that $x * y$ holds. For each positive integer i, denote by H_i the following two statements.

3.3) Any sequence of inputs $I_1, \cdots, I_j$, of length $j \leqq i$, which is applicable to x, is applicable to y and $\lambda_S(x, I_1, \cdots, I_j)=\lambda_Y(y, I_1, \cdots, I_j)$.

3.4) For any sequence of inputs $I_1, \cdots, I_j$, of length $j \leqq i$, and some input I, if $\delta_S(x, I_1, \cdots, I_j)$ and $\lambda_S(x, I_1, \cdots, I_jI)$ exist, then $\delta_Y(y, I_1, \cdots, I_j)$ exists and $\delta_S(x, I_1, \cdots, I_j) * \delta_Y(y, I_1, \cdots, I_j)$ holds.

From the definition of $x \leqq y$, in order to show that $x \leqq y$ it suffices to prove that H_i holds for each integer i.

Obviously, H_1 holds from the definition of $x * y$. Continuing by mathematical induction, assume that H_i is true for every positive integer $1 \leqq i \leqq k$. It will now be shown that H_{k+1} is true. Since H_k is true, in showing that both 3.3) and 3.4) for H_{k+1} are true, it suffices to consider only sequences of inputs $I_1, \cdots, I_{k+1}$ of length $k+1$.

Let $I_1, \cdots, I_{k+1}$ be any sequence of inputs, of length $k+1$, which is applicable to x. Since $I_1, \cdots, I_k$ is applicable to x and H_k is true,

$$\lambda_S(x, I_1, \cdots, I_k) = \lambda_Y(y, I_1, \cdots, I_k).$$

[1] A *binary relation* R between the states K_S of machine S and the states K_Y of machine Y is a subset $A(R)$ of the set of all ordered pairs $K_S \times K_Y$. For p in K_S and q in K_Y, pRq is said to *hold* if and only if (p, q) is in $A(R)$. Thus a binary relation R between K_S and K_Y is determined when the pairs (p, q), for which pRq hold are determined.

[2] A relation $*$ satisfying the hypothesis of Theorem 3 is hereafter called a *star relation between S and Y*.

Since $\lambda_S(x, I_1, \cdots, I_{k+1}) = \lambda_S[\delta_S(x, I_1, \cdots, I_k), I_{k+1}]$ exists and 3.4) for H_k holds, $\delta_S(x, I_1, \cdots, I_k)$ and $\delta_Y(y, I_1, \cdots, I_k)$ exist and

$$\delta_S(x, I_1, \cdots, I_k) * \delta_Y(y, I_1, \cdots, I_k).$$

Then

$$\begin{aligned}\lambda_S(x, I_1, \cdots, I_{k+1}) &= \lambda_S(x, I_1, \cdots, I_k)\lambda_S[\delta_S(x, I_1, \cdots, I_k), I_{k+1}] \\ &= \lambda_Y(y, I_1, \cdots, I_k)\lambda_Y[\delta_Y(I_1, \cdots, I_k), I_{k+1}] \\ &= \lambda_Y(y, I_1, \cdots, I_{k+1}).\end{aligned}$$

Thus (3.3) is true for H_{k+1}.

Now let $\delta_S(x, I_1, \cdots, I_{k+1})$ and $\lambda_S(x, I_1, \cdots, I_{k+1}, I)$ exist for a sequence of inputs $I_1, \cdots, I_{k+1}$ of length $k+1$ and some input I. Then $\delta_S(x, I_1, \cdots, I_k) * \delta_Y(y, I_1, \cdots, I_k)$ holds. Now

$$\lambda_S(x, I_1, \cdots, I_{k+1}, I) = \lambda_S(x, I_1, \cdots, I_k)\lambda_S[\delta_S(x, I_1, \cdots, I_k), I_{k+1}, I]$$

exists. Thus $\lambda_S[\delta_S(x, I_1, \cdots, I_k), I_{k+1}, I]$ exists. By (3.2)(b), $\delta_Y[\delta_Y(y, I_1, \cdots, I_k), I_{k+1}]$ exists and

$$\begin{aligned}&\delta_S(x, I, \cdots, I_{k+1}) \\ &= \delta_S[\delta_S(x, I_1, \cdots, I_k), I_{k+1}] * \delta_Y[\delta_Y(y, I_1, \cdots, I_k), I_{k+1}] \\ &= \delta_Y(y, I_1, \cdots, I_{k+1}).\end{aligned}$$

Hence 3.4) is true for H_{k+1}.

Since both 3.3) and 3.4) for H_{k+1} are true, H_{k+1} is true. By the principle of mathematical induction, H_i is true for every positive integer i. Hence $x \leqq y$ and the theorem is proved.

For a given star relation, it is possible for $x \leqq y$ to be true but $x * y$ to be false.

There may exist many star relations between two machines S and Y. Of course if $S \leqq Y$ holds, then writing $x * y$ if and only if $x \leqq y$ defines one particular star relation between S and Y.

Now consider the problem of executing steps 2)–3). The method given here consists of the following steps.

3.5) A machine Y having the number of states dictated by steps 2)–3) and such that $S \leqq Y$ is assumed to exist. However, the δ, λ matrix of Y is unknown.

3.6) A star relation between S and Y is assumed to exist. However, the pairs of states (x, y) for which $x * y$ holds are unknown.

3.7) Mathematical consequences of 3.5) and 3.6) are used to construct both the δ, λ matrix of Y and the pairs (x, y) for which $x * y$ holds.

Let us apply 3.5), 3.6), and 3.7) to machine S of Example 1. As shown in Section II, any machine Y with the property that $S \leqq Y$ has at least four states. Suppose that one such four-state machine Y exists. Label its states q_1, q_2, q_3, and q_4. In view of (3.1) and the fact that $\{p_2, p_3, p_4, p_6\}$ is a set of incompatible states, by a change of notation if necessary, it may be assumed that $p_2 * q_1$, $p_3 * q_2$, $p_4 * q_3$, and $p_6 * q_4$ all hold. Since $\{p_2, p_3, p_4, p_6\}$ is a set of incompatible states, it follows that

$$\begin{aligned}&p_2 \phi q_2,\ p_2 \phi q_3,\ p_3 \phi q_4, \\ &p_3 \phi q_1,\ p_3 \phi q_3,\ p_3 \phi q_4, \\ &p_4 \phi q_1,\ p_4 \phi q_2,\ p_4 \phi q_4,\end{aligned}$$

and

$$p_6 \phi q_1,\ p_6 \phi q_2,\ p_6 \phi q_3.$$

From $p_1 \phi p_4$, $p_1 \phi p_6$, $p_4 * q_3$, and $p_6 * q_4$, it follows that $p_1 \phi q_3$ and $p_1 \phi q_4$. From $p_5 \phi p_2$, $p_5 \phi p_3$, $p_2 * q_1$, and $p_3 * q_2$, it follows that $p_5 \phi q_1$ and $p_5 \phi q_2$.

At this moment there is associated with the star relation under construction: a) pairs $p * q$ which are presumed to be in the star relation, and hence are listed under the "Certain" column below; and b) pairs $p * q$ which, as yet, are presumed to be neither included nor excluded from the star relation, and hence are listed under the "Uncertain" column below. In the sequel, the listing of the pairs in both the Certain and Uncertain columns is labelled *tentative, incomplete star relation.*

Certain	*Uncertain*	
$p_2 * q_1$	$p_1 * q_1$	$p_1 * q_2$
$p_3 * q_2$		
$p_4 * q_3$	$p_5 * q_3$	$p_5 * q_4$
$p_6 * q_4$		

The following terminology is now introduced. If $x * y$ holds and if I is an input applicable to x, then $(x * y, I)$ is said to yield the two (easily verified) facts: $\lambda_Y(y, I) = \lambda_S(x, I)$; and $\delta_S(x, I) * \delta_Y(y, I)$. Moreover, if the only states z in Y such that $\delta_S(x, I) * z$ holds are $z_1, \cdots, z_t$, then $(x * y, I)$ is said to yield the (easily verified) fact $\delta_Y(y, I) = z_1$ or $\cdots$ or $\delta_Y(y, I) = z_t$. In particular, if there is only one such state, namely z_1, then $(x * y, I)$ is said to yield $\delta_Y(y, I) = z_1$.

$(p_2 * q_1, a)$ yields $q_4 = \delta(q_1, a)$ and $3 = \lambda(q_1, a)$.

$(p_2 * q_1, b)$ yields $q_4 = \delta(q_1, b)$ and $1 = \lambda(q_1, b)$.

$(p_3 * q_2, a)$ yields $q_3 = \delta(q_2, a)$ and $3 = \lambda(q_2, a)$.

$(p_3 * q_2, b)$ yields $q_3 = \delta(q_2, b)$ and $2 = \lambda(q_2, b)$.

$(p_4 * q_3, a)$ yields $q_1 = \delta(q_3, a)$ and $4 = \lambda(q_3, a)$.

$(p_4 * q_3, c)$ yields $q_1 = \delta(q_3, c)$ and $1 = \lambda(q_3, c)$.

$(p_6 * q_4, a)$ yields $q_2 = \delta(q_4, a)$ and $4 = \lambda(q_4, a)$.

The δ, λ facts currently known about machine Y are summarized in Fig. 3.

At this point, all "obvious" mathematical consequences of Y having just four states have been derived. To continue, a guessing procedure is instituted. In particular, a guess is made about either the star relation or the δ, λ matrix of Y. Mathematical consequences of this conjecture are then derived. Then another conjecture is made, etc. The guesses are made in such a way as to encompass all machines with four states ultimately. If, at any step, a contradiction arises, then the

	q_1	q_2	q_3	q_4
a	q_4 3	q_3 3	q_1 4	q_2 4
b	q_4 1	q_3 2		
c			q_1 1	

Fig. 3—Tentative, incomplete solution to Example 1.

most recent guess is changed. This, in turn, frequently causes changes to be made in earlier guesses. If in this way both a star relation and a machine Y are constructed, then a solution to Problem Q is found. Should all guesses in the sequence of guesses encompassing all machines with four states lead to contradictions, then no machine Y with four states, such that $S \leqq Y$ holds, exists.

Guess 1: $p_1 * q_1$ holds.

$(p_1 * q_1, a)$ yields $p_5 * q_4$ and $\lambda(q_1, a) = 3$.

$(p_1 * q_1, c)$ yields $\delta(q_1, c) = q_3$ and $\lambda(q_1, c) = 3$.

$(p_5 * q_4, a)$ yields $p_1 * \delta(q_4, a) = q_2$ and $\lambda(q_4, a) = 4$.

$(p_1 * q_2, a)$ yields $p_5 * \delta(q_2, a) = q_3$ and $\lambda(q_2, a) = 3$.

$(p_1 * q_2, c)$ yields $q_3 = \delta(q_2, c)$ and $\lambda(q_2, c) = 3$.

$(p_5 * q_3, a)$ yields $p_1 * q_1$ and $\delta(q_3, a) = 4$.

The results to date are summarized in Figs. 4 and 5.[3] They indicate that a machine Y and a star relation between S and Y have been constructed. Thus machine Y given in Fig. 4 is a solution to Problem Q for the first example.

A solution to Problem Q for Example 1 cannot be obtained by any other method known to the author. In particular, the well-known merging technique[4] applied to S yields S itself since, as is easily seen, no merges are possible.

IV. Other Examples

In this section, two additional examples are presented, and a solution to Problem Q for each of them is obtained. In addition, a slight modification of the synthesis technique, which lessens the amount of computation, is given.

[3] The numbers "1," "2," etc., either above or to the right of an item in a figure indicates that the derivation of that fact depended on guesses **1**, 2, etc.

[4] Given a machine S, let $\{Q_1, Q_2, \cdots, Q_r\}$ be a family of non-empty sets of states of S satisfying the following conditions.

a) Each state of S is in one and only one of the Q_i.

b) For each Q_i and for each two elements q_j and q_k in Q_i, if I is any input such that $\lambda(q_j, I)$ and $\lambda(q_k, I)$ exist, then $\lambda(q_j, I) = \lambda(q_k, I)$.

c) For each Q_i and for each two elements q_j and q_k in Q_i, if I is any input such that $\delta(q_j, I)$ and $\delta(q_k, I)$ exist, then $\delta(q_j, I)$ and $\delta(q_k, I)$ are in the same set Q_t.

Then a machine Y is obtained by the following process. The states of Y are the symbols $Q_1, \cdots, Q_r$. If $\delta(q, I)$ exists for some q in Q_i and $\delta(q, I)$ is in Q_j, write $\delta_Y(Q_i, I) = Q_j$. If $\lambda(q, I)$ exists for some q in Q_i, write $\lambda_Y(Q_i, I) = \lambda(q, I)$. The machine thus defined is said to be the machine resulting from the merging of the states comprising $Q_1, Q_2, \cdots, Q_r$. The entire procedure is called the *merging technique*. See [1], [3]–[5].

	q_1	q_2	q_3	q_4
a	q_4 3	q_3 3	q_1 4	q_2 4
b	q_4 1	q_3 2		
c	$\frac{1}{q_3}$ $\frac{1}{3}$	$\frac{1}{q_3}$ $\frac{1}{3}$	q_1 1	

Fig. 4—A Solution Y to Example 1.

$p_2 * q_1$	$p_1 * q_1$; 1
$p_3 * q_2$	$p_5 * q_4$; 1
$p_4 * q_3$	$p_1 * q_2$; 1
$p_6 * q_4$	$p_5 * q_3$; 1

Fig. 5—The star relation for Example 1.

For the second example, let S be the machine whose δ, λ matrix is given in Fig. 6. Then

4.1) $P_1 = H \cup \{(p_1, p_2), (p_1, p_4), (p_1, p_5), (p_2, p_3), (p_2, p_5), (p_2, p_6), (p_2, p_7), (p_2, p_8), (p_3, p_4), (p_3, p_6), (p_3, p_8), (p_4, p_5), (p_4, p_7), (p_6, p_8)\}$,

where

$$H = \{(p_i, p_i) \mid 1 \leqq i \leqq 8\},$$

4.2) $P_2 = H \cup \{(p_1, p_2), (p_1, p_4), (p_1, p_5), (p_2, p_3), (p_2, p_5), (p_2, p_6), (p_2, p_7), (p_2, p_8), (p_3, p_4), (p_3, p_8), (p_4, p_5), (p_4, p_7), (p_6, p_8)\}$,

and

$$P_3 = P_2.$$

Therefore, the compatible pairs of states of S are precisely those pairs in P_2. Thus the incompatible pairs are

4.3) $(p_1, p_3), (p_1, p_6), (p_1, p_7), (p_1, p_8), (p_2, p_4), (p_3, p_5), (p_3, p_6), (p_3, p_7), (p_4, p_6), (p_4, p_8), (p_5, p_6), (p_5, p_7), (p_5, p_8), (p_6, p_7), (p_7, p_8)$.

If the reader agrees that $\{p_1, p_3, p_6, p_7\}$ is a numerically maximal set of incompatible states of S, then he may skip the table given below. If not, the method for obtaining a numerically maximal set of incompatible states yields the following:

Old A	*Old B*	*Steps*	*Comments*
$\{p_1\}$	$\{p_1\}$	2.6), 2.7), 2.8)a)	$C(p_1) = \{p_3, p_6, p_7, p_8\}$
$\{p_1\}$	$\{p_1, p_3\}$	2.7), 2.8)a)	$C(p_3) = \{p_6, p_7\}$
$\{p_1\}$	$\{p_1, p_3, p_6\}$	2.7), 2.8)a)	$C(p_6) = \{p_7\}$
$\{p_1\}$	$\{p_1, p_3, p_6, p_7\}$	2.7), 2.8)b)	$C(p_7)$ is empty
$\{p_1, p_3, p_6, p_7\}$	$\{p_1, p_3, p_6, p_7\}$	2.9)b)	no $C(p_i)$ has more than four states
$\{p_1, p_3, p_6, p_7\}$	$\{p_2\}$	2.7), 2.8)c), 2.9)b)	$C(p_2) = \{p_4\}$
$\{p_1, p_3, p_6, p_7\}$	$\{p_3\}$	2.7), 2.8)c), 2.9)b)	$C(p_3) = \{p_5, p_6, p_7\}$
$\{p_1, p_3, p_6, p_7\}$	$\{p_4\}$	2.7), 2.8)c), 2.9)b)	$C(p_4) = \{p_6, p_8\}$
$\{p_1, p_3, p_6, p_7\}$	$\{p_5\}$	2.7), 2.8)c), 2.9)b)	$C(p_5) = \{p_6, p_7, p_8\}$
$\{p_1, p_3, p_6, p_7\}$	$\{p_6\}$	2.7), 2.8)c), 2.9)b)	$C(p_6) = \{p_7\}$
$\{p_1, p_3, p_6, p_7\}$	$\{p_7\}$	2.7), 2.8)c), 2.9)b)	$C(p_7) = \{p_8\}$
$\{p_1, p_3, p_6, p_7\}$	$\{p_8\}$	2.7), 2.8)c), 2.9)b)	$C(p_8)$ is empty; the process terminates.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
a	p_2 1		p_7 2		p_5 1		p_6 2	
b		p_4 2		p_2 1		p_1 2		p_4 2
c	p_3 2		p_6 1		p_3 2	p_7 1	p_1 2	p_2 1

Fig. 6—Machine S of Example 2.

From the above table, $\{p_1, p_3, p_6, p_7\}$ is a numerically maximal set of incompatible states of S.

Now suppose that there exists a four-state machine Y such that $S \leqq Y$. Label the states of Y as q_1, q_2, q_3, and q_4. Without loss of generality, it may be assumed that $p_1 * q_1$, $p_3 * q_2$, $p_6 * q_3$, and $p_7 * q_4$ all hold. Since $\{p_1, p_3, p_6, p_7\}$ is a set of incompatible states

$$p_1 \phi q_2,\ p_1 \phi q_3,\ p_1 \phi q_4,$$
$$p_3 \phi q_1,\ p_3 \phi q_3,\ p_3 \phi q_4,$$
$$p_6 \phi q_1,\ p_6 \phi q_2,\ p_6 \phi q_4,$$
$$p_7 \phi q_1,\ p_7 \phi q_2,\ p_7 \phi q_3,$$

Since p_2 is incompatible only with p_4, $p_2 * q_1$, $p_2 * q_2$, $p_2 * q_3$, and $p_2 * q_4$ are possible.

Since $p_4 \phi p_6$, $p_4 \phi q_3$. Thus $p_4 * q_1$, $p_4 * q_2$, and $p_4 * q_4$ are possible.

Since $p_5 \phi p_3$, $p_5 \phi p_6$, and $p_5 \phi p_7$; $p_5 \phi q_2$, $p_5 \phi q_3$, and $p_5 \phi q_4$. Thus $p_5 * q_1$ holds.

Since $p_8 \phi p_1$ and $p_8 \phi p_7$, $p_8 \phi q_1$ and $p_8 \phi q_4$. Thus $p_8 * q_2$ and $p_8 * q_3$ are possible.

$(p_1 * q_1, c)$ yields $\delta(q_1, c) = q_2$ and $\lambda(p_1, c) = 2$.

$(p_3 * q_2, a)$ yields $\delta(q_2, a) = q_4$ and $\lambda(q_2, a) = 2$.

$(p_3 * q_2, c)$ yields $\delta(q_2, c) = q_3$ and $\lambda(q_2, c) = 1$.

$(p_5 * q_1, a)$ yields $\delta(q_1, a) = q_1$ and $\lambda(q_1, a) = 1$.

$(p_5 * q_1, c)$ yields $\delta(q_1, c) = q_2$ and $\lambda(q_1, c) = 2$.

$(p_6 * q_3, b)$ yields $\delta(q_3, b) = q_1$ and $\lambda(q_3, b) = 2$.

$(p_6 * q_3, c)$ yields $\delta(q_3, c) = q_4$ and $\lambda(q_3, c) = 1$.

$(p_7 * q_4, a)$ yields $\delta(q_4, a) = q_3$ and $\lambda(q_4, a) = 2$.

$(p_7 * q_4, c)$ yields $\delta(q_4, c) = q_1$ and $\lambda(q_4, c) = 2$.

$(p_1 * q_1, a)$ yields $p_2 * \delta(q_1, a) = q_1$ and $\lambda(q_1, a) = 1$.

Since $p_4 \phi p_2$, $p_4 \phi q_1$.

The results to date are summarized in Figs. 7 and 8.

At this point, the guessing procedure is initiated. In order to reduce the number of conjectures, it seems reasonable to focus attention on those states p for which there is no definite state q such that $p * q$ holds. With both p_4 and p_8 are associated two states. Since there is no reason to favor either p_4 or p_8, as would be the case if fewer states were associated with one of them, the state with the smaller index, in this case p_4 ,is first considered.

	q_1	q_2	q_3	q_4
a	q_1 1	q_4 2		q_3 2
b			q_1 2	
c	q_2 2	q_3 1	q_4 1	q_1 2

Fig. 7—Tentative, incomplete solution to Example 2.

Certain	*Uncertain*		
$p_1 * q_1$	$p_2 * q_2$	$p_2 * q_3$	$p_2 * q_4$
$p_2 * q_1$			
$p_3 * q_2$	$p_4 * q_2$	$p_4 * q_4$	
$p_5 * q_1$	$p_8 * q_2$	$p_8 * q_3$	
$p_6 * q_3$			
$p_7 * q_4$			

Fig. 8—Tentative, incomplete star relation for Example 2.

Guess 1: $p_4 * q_2$ holds.

Since $p_8 \phi p_4$, $p_8 \phi q_2$. Thus $p_8 * q_3$ holds.

$(p_8 * q_3, b)$yields $p_4 * \delta(q_3, b) = q_1$.

This is impossible. Hence, guess 1 is incorrect, and $p_4 * q_2$ does not hold.

Guess 2: $p_4 * q_4$ holds.

Since $p_2 \phi p_4$, $p_2 \phi q_4$.

$(p_2 * q_1, b)$ yields $\delta(q_1, b) = q_4$ and $\lambda(q_1, b) = 2$.

Guess 3: $p_8 * q_2$ holds.

$(p_8 * q_2, b)$ yields $\delta(q_2, b) = q_4$ and $\lambda(q_2, b) = 2$.

$(p_8 * q_2, c)$ yields $p_2 * \delta(q_2, c) = q_3$ and $\lambda(q_2, c) = 1$.

$(p_2 * q_3, b)$ yields $q_4 = \delta(q_3, b)$, a contradiction.

Thus $p_8 * q_2$ does not hold.

Guess 4: $p_8 * q_3$ holds.

$(p_8 * q_3, b)$ yields $p_4 * \delta(q_3, b) = q_1$, a contradiction.

Therefore guess 4 is incorrect; thus, guess 2 is also incorrect. Consequently, there is no four-state machine Y such that $S \leqq Y$ holds.

The reader may have observed that in both Example 1 and that portion of Example 2 already considered, no contradictions arose from the output function, that is, from $\lambda(q, I) = E$ through one channel of reasoning and

$\lambda(q, I) \neq E$ through another channel.[5] This was no accident due to the particular examples chosen, but was due to the manner in which the alternatives were enumerated and then selected, as is now shown.

Consider the output $\lambda(q, I)$ for a state q and an input I. Each output $\lambda(q, I)$ is determined in only two ways, namely: a) $\lambda(q, I) = \lambda(p, I)$ for some $p * q$ already in the star relation; and b) $\lambda(q, I)$ is defined arbitrarily (thus making a guess about the nature of those states p for which $p * q$ holds). Suppose, as was done in Example 1 and that portion of Example 2 already considered, that only a) is to be used in the technique.

4.1) Suppose that a particular $p_i * q_j$ is inserted into the star relation under construction.

4.2) Let $p_{i(1)}, p_{i(2)}, \cdots, p_{i(r)}$ be all the states which are incompatible with p_i.

4.3) Then $p_{i(1)} \phi q_j, p_{i(2)} \phi q_j, \cdots$, and $p_{i(r)} \phi q_j$ occur.

4.4) Remove from the Uncertain column those $p * q$ among

$$p_{i(1)} * q_j, p_{i(2)} * q_j, \cdots, \text{and } p_{i(r)} * q_j$$

which are in that column.

Suppose that some $p_a * q_j$ is now selected from the Uncertain column and inserted into the star relation. Suppose further that for some input I, $(p_i * q_j, I)$ yields $\lambda(q_j, I) = E$ and $(p_a * q_j, I)$ yields $\lambda(q_j, I) \neq E$, where $p_i * q_j$ is already in the star relation. Then $\lambda(p_i, I) = E$ and $\lambda(p_a, I) \neq E$. Therefore p_a is a state which is incompatible with p_i. From 4.2), 4.3), and 4.4), $p_a * q_j$ cannot appear in the Uncertain column. From this impossibility, it follows that if 4.2), 4.3), and 4.4) are executed for each $p_i * q_j$ as it is inserted into the star relation, and if no arbitrary guesses about the output $\lambda(q, I)$ are made, then no contradiction of a guess can arise through consideration of the λ function. If no contradiction of a guess can arise through consideration of the λ function, then the star relation and the δ function of Y can be constructed first, with the λ function then being constructed automatically from the star relation. If this modified procedure is followed, it usually results in less computation than the earlier procedure since the output $\lambda(q, I)$ need not be calculated for incorrect guesses. Another advantage of this modified procedure is to decrease that portion of the reduction technique which is not automatic. Because of these two advantages, the modified procedure is pursued in the sequel.

Returning to Example 2, suppose that there exists a five-state machine Y such that $S \leqq Y$ holds. It is easily seen that the merging of p_1 and p_4 to Q_1, p_2 and p_7 to Q_2, p_3 to Q_3, p_5 to Q_4, and p_6 and p_8 to Q_5 yields the five-state machine Y given in Fig. 9. For this Y, $S \leqq Y$ holds so that a solution to Problem Q for Example 2 has been found. If, for some reason, one does not wish to use the merging technique, then a five-state machine Y such that $S \leqq Y$ can be found as follows.

[5] This was pointed out to the author by C. C. Chang.

	Q_1	Q_2	Q_3	Q_4	Q_5
a	Q_2 1	Q_5 2	Q_2 2	Q_4 1	
b	Q_2 1	Q_1 2			Q_1 2
c	Q_3 2	Q_1 2	Q_5 1	Q_3 2	Q_2 1

Fig. 9—Solution to Example 2 obtained by the merging technique.

Let Y be a five-state machine which is a solution to Problem Q. Label the states of Y as q_1, q_2, q_3, q_4, and q_5. Without loss of generality, it may be assumed that $p_1 * q_1$, $p_3 * q_2$, $p_6 * q_3$, and $p_7 * q_4$ hold. As before, $p_1 \phi q_2$, $p_1 \phi q_3$, and $p_1 \phi q_4$, so that $p_1 * q_5$ might hold. Similarly, $p_3 * q_5$, $p_6 * q_5$, $p_7 * q_5$, $p_2 * q_1$, $p_2 * q_2$, $p_2 * q_3$, $p_2 * q_4$, $p_2 * q_5$, $p_4 * q_1$, $p_4 * q_2$, $p_4 * q_4$, $p_4 * q_5$, $p_5 * q_1$, $p_5 * q_5$, $p_8 * q_2$, $p_8 * q_3$, and $p_8 * q_5$ might hold.

At this point, the guessing procedure is initiated. In general, it is desirable to construct a star relation having many states p such that $p * q$ holds for just one state q. For if $p * q$ holds for just one state q, then whenever $q * \delta(x, I)$ holds, it follows that $q = \delta(x, I)$. In Example 2, one way to obtain states p such that $p * q$ holds for just one state q is to assume that one of the states in the numerically maximal set of incompatible states is in the star relation with q_5. (It turns out that a star relation also exists such that $p * q_5$ does not hold when $p = p_1, p_3, p_6$, and p_7.) This is now done.

Guess 5: $p_1 * q_5$ holds.

Since $p_3 \phi p_1$, $p_6 \phi p_1$, $p_7 \phi p_1$ and $p_8 \phi p_1$, it follows that $p_3 \phi q_5$, $p_6 \phi q_5$, $p_7 \phi q_5$, and $p_8 \phi q_5$.

$(p_1 * q_1, a)$ yields $p_2 * \delta(q_1, a)$.

$(p_1 * q_1, c)$ yields $\delta(q_1, c) = q_2$.

$(p_3 * q_2, a)$ yields $\delta(q_2, a) = q_4$.

$(p_3 * q_2, c)$ yields $\delta(q_2, c) = q_3$.

$(p_6 * q_3, b)$ yields $\delta(q_3, b) = q_1$ or q_5.

$(p_6 * q_3, c)$ yields $\delta(q_3, c) = q_4$.

$(p_7 * q_4, a)$ yields $\delta(q_4, a) = q_3$.

$(p_7 * q_4, c)$ yields $\delta(q_4, c) = q_1$ or q_5.

$(p_1 * q_5, a)$ yields $p_2 * \delta(q_5, a)$.

$(p_1 * q_5, c)$ yields $\delta(q_5, c) = q_2$.

The criterion mentioned prior to Guess 1 of Example 2 is now applied.

Guess 6: $p_5 * q_1$ holds.

$(p_5 * q_1, a)$ yields $p_5 * \delta(q_1, a)$.

$(p_5 * q_1, c)$ yields $\delta(q_1, c) = q_2$.

	q_1	q_2	q_3	q_4	q_5
a	7 / q_1	5 / q_4		5 / q_3	
b			5 / q_1 or 5 / q_5		
c	5 / q_2	5 / q_3	5 / q_4	5 / q_1 or 5 / q_5	5 / q_2

Fig. 10—Tentative, incomplete solution to Example 2.

Certain	*Uncertain*
$p_1 * q_1$	~~$p_1 * q_5$~~; 5
$p_3 * q_2$	~~$p_2 * q_1$~~; 7 $p_2 * q_2$; $p_2 * q_3$; $p_2 * q_4$; ~~$p_2 * q_5$~~; 7
$p_6 * q_3$	~~$p_3 * q_5$~~; 5
$p_7 * q_4$	~~$p_4 * q_1$~~; 7 $p_4 * q_2$; $p_4 * q_4$; $p_4 * q_5$
$p_1 * q_5$; 5	~~$p_5 * q_1$~~; 6 $p_5 * q_5$
~~$p_2 * \delta(q_1, a)$~~; 5, 7	~~$p_6 * q_5$~~; 5
$p_2 * \delta(q_5, a)$; 5	~~$p_7 * q_5$~~; 5
$p_5 * q_1$; 6	$p_8 * q_2$; $p_8 * q_3$; ~~$p_8 * q_5$~~; 5
~~$p_5 * \delta(q_1, a)$~~; 6, 7	
$p_2 * q_1$; 7	
$p_4 * \delta(q_1, b)$; 7	

Fig. 11—Tentative, incomplete star relation for Example 2.

	q_1	q_2	q_3	q_4	q_5
a	q_1	q_4		q_3	q_1
b	q_5	q_5	q_5		q_1
c	q_2	q_3	q_4	q_1	q_2

Fig. 12—Incomplete solution to Example 2.

$p_1 * q_1$ $p_5 * q_1$
$p_1 * q_5$
$p_6 * q_3$
$p_2 * q_1$
$p_2 * q_3$ $p_7 * q_4$
$p_3 * q_2$ $p_8 * q_2$
$p_4 * q_5$

Fig. 13—Star relation for Example 2.

	q_1	q_2	q_3	q_4	q_5
a	q_1 1	q_4 2		q_3 2	q_1 1
b	q_5 2	q_5 2	q_5 2		q_1 1
c	q_2 2	q_3 1	q_4 1	q_1 2	q_2 2

Fig. 14—Solution to Example 2.

Since $p_5 * \delta(q_1, a)$ holds, either $\delta(q_1, a) = q_1$ or $\delta(q_1, a) = q_5$. Since $p_2 * \delta(q_1, a)$ holds, either $p_2 * q_1$, or $p_2 * q_5$ holds.

Guess 7: $p_2 * q_5$ does not hold. Then $p_2 * q_1$ holds and $\delta(q_1, a) = q_1$.

Since $p_4 \phi p_2$, $p_4 \phi q_1$.

$(p_2 * q_1, b)$ yields $p_4 * \delta(q_1, b)$.

The results to date are summarized in Figs. 10 and 11.

The criterion given prior to Guess 1 of Example 2 is applied again.

Guess 8: $p_8 * q_2$ holds.

Since $p_4 \phi p_8$, $p_4 \phi q_2$.

$(p_8 * q_2, b)$ yields $p_4 * \delta(q_2, b)$.

$(p_8 * q_2, c)$ yields $p_2 * q_3$.

$(p_2 * q_3, b)$ yields $p_4 * \delta(q_3, b) = q_1$ or q_5.

Since $p_4 * q_1$ does not hold, $p_4 * q_5$ holds and $\delta(q_3, b) = q_5$.

$(p_4 * q_5, b)$ yields $p_2 * \delta(q_5, b)$.

The remaining guesses are made to minimize the number of pairs in the star relation. Thus, since $p_2 * \delta(q_5, a)$ and $p_2 * q_1$,

Guess 9: $\delta(q_5, a) = q_1$.

Since $p_2 * \delta(q_5, b)$ and $p_2 * q_1$,

Guess 10: $\delta(q_5, b) = q_1$.

Since $p_4 * \delta(q_1, b)$ and $p_4 * q_5$,

Guess 11: $\delta(q_1, b) = q_5$.

Since $p_4 * \delta(q_2, b)$ and $p_4 * q_5$,

Guess 12: $\delta(q_2, b) = q_5$.

Finally

Guess 13: $\delta(q_4, c) = q_1$.

At this point, a star relation and the δ function for Y are constructed. These are given in Figs. 12 and 13, respectively. From $(p_1 * q_1, a)$ yielding $\lambda(q_1, a) = 1$, $(p_1 * q_1, c)$ yielding $\lambda(q_1, c) = 2$, etc., the λ function for Y is constructed. The δ, λ matrix for Y, that is, a solution to Problem Q for Example 2, is given in Fig. 14.

It is a fact that the machine Y just obtained cannot be derived by the merging technique (although a solution to Problem Q for Example 2, as already noted, can be obtained by the merging technique). Suppose that S can be merged to form Y. Since p_3 and q_2 are the only states yielding outputs of 2 and 1 when inputs a and c are applied, p_3 must be merged to q_2. Then $\delta(p_3, a) = p_7$ must be merged to $\delta(q_2, a) = q_4$ and $\delta(p_3, c) = p_6$ to $\delta(q_2, c) = q_3$. Then $\delta(p_6, b) = p_1$ must be merged to $\delta(q_3, b) = q_5$ and $\delta(p_7, c) = p_1$ to $\delta(q_4, c) = q_1$. But p_1 cannot be merged simultaneously to two distinct states. Hence S cannot be merged to form Y.

Each of the machines S in Examples 1 and 2 has the property, that $\delta(q, I)$ and $\lambda(q, I)$ exist simultaneously.

	p_1	p_2	p_3	p_4	p_5	p_6
a	1		1		p_2 2	
b		p_5 2		1		p_1 1
c	p_6 3		p_4	p_3	1	2

Fig. 15—Machine S of Example 3.

	q_1	q_2	q_3
a	1	q_1 2	
b	q_2 2		q_1 1
c	q_3 3	1	q_1 2

Fig. 16—Solution to Example 3.

However the technique presented is applicable even if S does not have the above property. For (the third) example, let S be the machine whose δ, λ matrix is given in Fig. 15. The incompatible pairs of states of S are

$$(p_1, p_5), (p_1, p_6), (p_2, p_4),$$
$$(p_2, p_6), (p_3, p_5), (p_5, p_6).$$

To find a three-state machine Y, with states q_1, q_2, and q_3, such that $S \leqq Y$, let $p_1 * q_1$, $p_5 * q_2$, and $p_6 * q_3$ hold. The three guesses $\delta(q_2, a) = q_1$, $p_3 * q_1$, and $\delta(q_3, c) = q_1$ determine a machine Y, exhibited in Fig. 16, and a star relation, exhibited in Fig. 17. The machine Y so constructed can also be obtained by merging p_1, p_2, and p_3 to q_1; p_5 to q_2; and p_4 and p_6 to q_3.

For some applications, *e.g.*, some switching problems, the only machines considered are those with the property that $\lambda(p, I) = \lambda(p, I^*)$ for each state p and each two inputs I and I^*, that is, the output is independent of the present input. The technique described in this paper is applicable to machines with the independence of present input property. In particular, suppose a given machine S which has the independence of present input property. For each state p, let I_p be an input such that $\lambda_S(p, I_p)$ is defined. Extend S by defining $\lambda_S(p, I) = \lambda_S(p, I_p)$ for all states p and all inputs I. As noted in the preceding paragraph, a solution T to Problem Q for the extended machine S may be found by the technique described in this paper. As is easily seen, machine T is also a solution to Problem Q for the original machine S and has the additional property that $\lambda_T(q, I)$ is independent of I.

$p_1 * q_1$
$p_5 * q_2$
$p_6 * q_3$
$p_2 * q_1$
$p_3 * q_1$
$p_4 * q_3$

Fig. 17—Star relation for Example 3.

Conclusion

A technique has been presented which, in theory at least, enables one to reduce a given machine to one having the fewest number of states possible. Part of the technique is automatic and part is not, the latter depending on the enumerating and then the selecting of different alternative paths of procedure. Because of this, the method as it now stands cannot be programmed on a computer. It is felt that through experience and further study, enough alternative paths of procedure and criteria for deciding which of the alternatives to select will arise so as to render the entire process automatic.

Bibliography

[1] D. D. Aufenkamp, "Analysis of sequential machines, II," IRE Trans. on Electronic Computers, vol. EC-7, pp. 299–306; December, 1958.
[2] S. Ginsburg, "On the reduction of superfluous states in a sequential machine," *J. Assoc. Comp. Mach.*, vol. 6, pp. 259–282; April, 1959.
[3] D. A. Huffman, "The synthesis of sequential switching circuits," *J. Franklin Inst.*, vol. 257, pp. 161–190, 275–303; March/April, 1954.
[4] G. H. Mealy, "A method for synthesizing sequential circuits," *Bell Sys. Tech. J.*, vol. 34, pp. 1045–1079; September, 1955.
[5] E. F. Moore, "Gedaken-experiments on sequential machines," in "Automata Studies," C. Shannon and J. McCarthy, Eds., Princeton University Press, Princeton, N. J., pp. 129–153; 1956.

Minimizing the Number of States in Incompletely Specified Sequential Switching Functions*

M. C. PAULL† AND S. H. UNGER†

Summary—Given a sequential switching function in the form of a flow table in which some of the entries are unspecified, the problem of reducing the number of rows in that flow table is extremely complex, and cannot, in general, be solved by any simple extension of the methods used for completely specified functions. An analysis of the problem is presented, and a partially enumerative solution is evolved. A rough indication of the efficiency of the given procedures may be obtained from the fact that these techniques have been successfully applied to approximately two dozen tables ranging up to about 15 rows. No solution required more than two hours.

I. Introduction

TERMINAL characteristics of sequential switching circuits (or machines) are conveniently specified in the form of flow tables[1–4] such as Table I. The rows represent *internal states*, and the columns represent *inputs*. For each *total state* (specified by both the internal state and the input), the table indicates the next internal state, and an output e_k. Thus, for example, if the system starts in state 3 and if the input sequence I_2, I_3, I_3, I_1, I_1, I_2 is applied, then the output sequences e_3, e_2, e_2, e_2, e_1, e_3 will result, and the final internal state will be 5. (This particular table represents a pulsed signal system.)

Previous work in this area has been applicable to machines that are synchronous or asynchronous, with pulsed signals or level type signals. We have found no reason to believe that the results presented here are any less general.

A flow table P will be said to *cover* a flow table Q (written as $P \geq Q$) if, for every state q of Q, there is a corresponding state p of P, such that for every possible sequence of inputs, the resulting output sequence from P when it starts in p will be identical to the sequence obtained from Q when it starts in q.[1–5] Given a fully specified flow table Q, an important problem in sequential circuit synthesis is to find the minimum-row member of the set of tables that cover Q. This minimization problem has been completely solved by Huffman, Moore, and Mealy.[1–3] Another paper on this subject has been written by Aufenkamp and Hohn.[6]

TABLE I

	I_1	I_2	I_3
1	1, e_1	5, e_3	1, e_4
2	3, e_2	4, e_3	2, e_2
3	1, e_1	2, e_3	3, e_1
4	5, e_1	4, e_3	3, e_1
5	2, e_1	5, e_1	2, e_1

The class of flow tables considered may be broadened to include unspecified or "don't care" entries. That is, for certain total states, the next-state entries or the outputs (or both) may be left unspecified. A detailed discussion of how such entries should be interpreted will be presented in Section II. The definition of covering can be generalized to include incompletely specified functions, and the minimization problem can then be reformulated in terms of the broader definition. This will be done in Section III.

Until recently, it was widely believed that the minimization of incompletely specified tables could always be accomplished with modified versions of the above mentioned methods.[7] However, Ginsburg[8–10] has recently discovered that there are examples of such flow tables that cannot be minimized through the use of the classical procedures. He has made a preliminary study of the situation and described some minimization procedures.

In this report, we shall present what we feel is a more thorough analysis of the problem as well as some minimization methods superior to those given by Ginsburg, although still not fully satisfactory in that some enumeration is required.

Formal proofs of theorems will be relegated to an Appendix in order to avoid interrupting the general development. A number of illustrative examples are given in Section V.

II. Incompletely Specified Functions

There are several ways of interpreting unspecified entries in flow tables. According to one interpretation, each blank entry may be filled arbitrarily, but a definite

* Manuscript received by the PGEC, April 14, 1959; revised manuscript received, July 1, 1959.

† Bell Telephone Labs., Inc., Whippany, N. J.

[1] D. A. Huffman, "The synthesis of sequential switching circuits," *J. Franklin Inst.*, vol. 257, pp. 161–190, 275–303; March/April, 1954.

[2] G. H. Mealy, "A method for synthesizing sequential circuits," *Bell Sys. Tech. J.*, vol. 34, pp. 1045–1079; September, 1955.

[3] E. F. Moore, "Gedanken experiments on sequential machines," in "Automata Studies," Princeton University Press, Princeton, N. J., pp. 129–153; 1956.

[4] S. H. Caldwell, "Switching Circuits and Logical Design," John Wiley and Sons, Inc., New York, N. Y.; pp. 453–659; 1958.

[5] This is the same as the equivalence definition used by Huffman, Moore, and Mealy for fully specified flow tables.

[6] D. D. Aufenkamp and F. E. Hohn, "Analysis of sequential machines," IRE Trans. on Electronic Computers, vol. EC-6, pp. 276–285; December, 1957.

[7] D. D. Aufenkamp, "Analysis of sequential machines, II," IRE Trans. on Electronic Computers, vol. EC-7, pp. 299–306; December, 1958.

[8] S. Ginsburg, "A Synthesis Technique for Minimal State Sequential Machines," Natl. Cash Register Co.; July, 1958. (Unpublished.)

[9] S. Ginsburg, "On the reduction of superfluous states in a sequential machine," *J. Assoc. Comp. Mach.*, vol. 6, pp. 259–282; April, 1959.

[10] S. Ginsburg, "A technique for the reduction of a given machine to a minimal-state machine," this issue, p. 346.

TABLE II

	I_1	I_2
1	1, –	2, e_1
2	3, e_1	1, e_1
3	2, e_2	1, e_1

(a)

	I_1	I_2
1	1, e_1	2, e_1
2	3, e_1	1, e_1
3	2, e_2	1, e_1

(b)

	I_1	I_2
1	1, e_2	2, e_1
2	3, e_1	1, e_1
3	2, e_2	1, e_1

(c)

	I_1	I_2
A	B, e_1	A, e_1
B	A, e_2	A, e_1

(d)

entry must be made. In other words, a function M, such as the one described by Table II(a), is covered by the functions described by both Table II(b) and Table II(c).

Another approach is to specify that the output will simply be ignored whenever the system enters a total state where the output is unspecified. The distinction between the two cases is that in the latter instance, it is not required that the *same* output always occur for the unspecified condition. That is, a function will be said to cover that of Table II(a) if at all times its behavior corresponds to *either* Table II(b) *or* Table II(c), but not necessarily always to the same one.

The difference between these two interpretations is quite significant when an attempt is made to minimize a table such as Table II(a). Applying classical reduction procedures, we see that neither Table II(b) nor Table II(c) can be reduced. Hence, if the unspecified entry is interpreted according to our first proposal, then Table II(a) cannot be reduced. However, under the second interpretation, Table II(d) covers Table II(a) and thus constitutes a reduced version of it. Row A corresponds to a combination of states 1 and 2 of the original table, and row B corresponds to a combination of states 1 and 3. This example constitutes a case where the classical procedures will not yield the best solution due to the implied interpretation of the unspecified entries.

Actually, the second interpretation that allowed us to reduce Table II(a) is the one that conforms to most situations in which such entries occur (for example, where certain input sequences are disallowed). Such a statement is not susceptible to proof, so it will simply be arbitrarily stated here that we shall use the second meaning. The aptness of this choice must be decided for each situation.

Although the illustrations in the preceding paragraph concerned output entries, the same arguments apply to unspecified next-state entries such as in Table III(a).

Using the idea that the desired machine need not behave as though some constant next-state entry were in place of the dash, Table III(a) can be reduced to Table III(b). Rows 1 and 2 are mapped into A, rows 1 and 3 are mapped into B, and row 4 is mapped into C. If it is required that the dash be replaced by some fixed-row number, then no reduction will be possible.

In the case of unspecified next-state entries, an equivalent way of stating our interpretation is to assume that no further inputs will occur after the machine enters the unspecified state.

A special comment is called for by the work of Aufenkamp.[7] He apparently accepts the same interpretation of incompletely specified functions that we are using, although he does not consider total states in which *only* the outputs are unspecified or in which *only* the next-states are unspecified.[11] It is therefore impossible to apply the procedures he describes to either of our two preceding examples. (Indeed, it does not seem possible even to *state* the second example in terms of Aufenkamp's connection matrices.)

TABLE III

	I_1	I_2
1	–, e_1	1, e_1
2	1, e_1	3, e_1
3	4, e_1	2, e_1
4	4, e_2	2, e_1

(a)

	I_1	I_2
A	A, e_1	B, e_1
B	C, e_1	A, e_1
C	C, e_2	A, e_1

(b)

TABLE IV

	I_1	I_2	I_3
1	–	1, 0	2, 0
2	1, 0	3, 0	2, 0
3	3, 1	2, 0	1, 0

(a)

	I_1	I_2	I_3	
A	A, 0	B, 0	A, 0	(12)
B	B, 1	A, 0	A, 0	(13)

(b)

We therefore present Flow Table IV(a), which is of the type which Aufenkamp claims to be able to reduce to its "simplest compatible form." Using the procedures to be described in the following sections, it is easy to reduce the 3-row Table IV(a) to the 2-row Table IV(b). As we understand Aufenkamp's procedures, they would fail to achieve any reduction, since he seeks only *disjoint* partitions of the states, whereas the given table can be reduced only by using the *overlapping* partition [(12), (13)]. This point will be amplified later.

Subsequent sections will concern the methods used in minimizing tables such as those discussed in the preceding examples.

III. General Nature of Minimization Process

An over-all picture of the minimization process will be presented in this section. Some definitions and a theorem are necessary prerequisites.

Definition 1: If a sequence of inputs is applied to flow table P (this is a less cumbersome, if less precise,

[11] *Ibid.;* Ginsburg *does* consider the case where only the next-state is unspecified, but, in effect, he does not distinguish between the case where both entries are blank and the case where only the output is unspecified.

reference to an input to *a machine described by* flow table P) when it is initially in state r, then this sequence will be said to be *applicable* to r if the state of the flow table is specified after each input, except possibly the last. That is, no unspecified next-state entries are encountered when an applicable input sequence is applied, except possibly at the final step.

Definition 2: A state p of a flow table P will be said to *cover* a state q of a flow table Q ($p \geq q$) if, when any input sequence applicable to q is applied to both P and Q when they are initially in states p and q respectively, the outputs obtained from P are the same as the outputs obtained from Q, wherever the latter are specified.

Definition 3: A flow table P *covers* a flow table Q ($P \geq Q$) if, for every state q in Q, there is a state p in P such that $p \geq q$.

Suppose that all the states of flow table Q can be grouped into sets $C_1, C_2, \cdots, C_m$ (C-sets), not necessarily disjoint, such that there is another flow table P with states $p_1, p_2, \cdots, p_m$, where p_1 covers all members of C_1, p_2 covers all members of C_2, and in general p_i covers all members of C_i ($i = 1, 2, \cdots, m$). It follows from definition 3 that $P \geq Q$, and if m is less than the number of states of Q, then P can be considered as a *reduced version* of Q. Our goal is to find a *minimum-row version* of the given table: a reduced version with the smallest number of rows.

Definition 4: Let $N(s, I_k)$ be the next-state entry (if specified) for state s and input I_k.

Definition 5: Let $Z(s, I_k)$ be the output (if specified) for state s and input I_k.

Definition 6: A set of states P is *implied* by a set of states R, if, for some input I_k, P is the set of all $N(r, I_k)$'s taken over all pairs (r, I_k), where r belongs to R and $N(r, I_k)$ is specified.

Definition 7: A grouping of all the rows of a flow table into C-sets (not necessarily disjoint) will be said to be *closed* if: 1) every set implied by any C_i is included in one of the C-sets; and 2) $Z(p, I_k) = Z(t, I_k)$ (if both are defined) where p and t belong to C_i, for every I_k and for every i.

Theorem 1: Every minimum-row version of a flow table corresponds to a closed grouping of C-sets of that table, in that each row of the reduced table covers all members of one C-set; and every closed collection of m C-sets corresponds to an m-row flow table that covers the given flow table, provided that every state of the given table is represented in at least one of the C-sets.

The following method always will be shown to yield a minimum-row flow table that covers a given n-row flow table Q.

Process A:

1) Group the states of Q into a number (less than n) of sets C_i which are not necessarily disjoint and whose union is the set of all states of Q.
2) Determine whether this grouping can correspond to a machine P which covers Q. (In other words, test the grouping for closure.)
3) Repeat this process for all possible groupings and select the grouping (or one of the groupings) with the smallest number of sets that satisfies step 2).
4) Form the reduced table (which is a minimum-row table according to Theorem 1), by using a single state in place of each set of states of the grouping found in step 3) as follows:
   a) $Z(p_i, I_k)$ is left unspecified if the outputs for all members of C_i are unspecified in column I_k of table Q. Any specified outputs in I_k for members of C_i will be the same (definition 7) and, in such cases, this is the value to use for $Z(p_i, I_k)$.
   b) $N(p_i, I_k)$ is left unspecified if next-state entries in column I_k are unspecified for all members of C_i. Otherwise (according to definiton 7), there will be at least one C-set such that all specified I_k next-state entries for C_i are included in that set. In this case, choose one such set, say C_j, and let $N(p_i, I_k) = p_j$.

It is important to realize that a state of Q may appear in more than one of these sets. This is the key point that was missed in applying the classical reduction process to incompletely specified machines. For example, the states of the machine described by Table III(a) are grouped as (12), (13), (4) to obtain Table III(b). The process of forming the C_i sets is not unique since, in general, many machines can be found to cover a given machine.

Process A is complete, and will always yield the desired result. However, except in the most trivial cases, an enormous amount of enumeration would be required in testing all groupings that satisfy step 1). Most of this enumeration can be eliminated, and the basis for a simplified approach will be discussed in the next section.

IV. Compatibles—Components of Groupings

Each grouping of states referred to in the previous section consists of a collection of sets of states. In this section, we shall describe procedures for determining which sets of states should be considered in such groupings.

Definition 8: A *compatible* is a set of states of a flow table that constitutes one member of some closed grouping. Every state of a flow table is, by itself, a compatible.

Definition 9: Two states are *incompatible* if they do not constitute a compatible.

One reduction procedure for flow table M may now be written as follows.

Process B:

1) If there is a compatible consisting of all states in M, then obviously it is a closed collection and thus corresponds to a reduced flow table.
2) If no one compatible corresponds to a reduced table, then test for closure (definition 7) each collection of compatibles whose union includes all

states of M, starting with 2-member groupings, then testing the 3-member groupings, etc. If no collection of $n-1$ or fewer compatibles is satisfactory, then M cannot be reduced.

3) If a satisfactory set of compatibles is found, then construct the reduced machine using step 4 of process A.

The next problem is to find the compatibles. One method is based on Theorem 2.

Theorem 2: A set of states is a compatible if, and only if, every pair of states in that set is a compatible.

Thus, by considering only pairs of states, all compatibles can be found. For example, if a, b, c and d are states, then abc is a compatible if ab, ac and bc are compatibles. If they are, then to see whether d can be added to this compatible it is necessary to determine whether ad, bd, and cd are also compatibles.

The problem now is to determine which pairs of rows of a given flow table are compatibles.

Definition 10: The *chain* generated by a set of states B is the collection of *all* sets produced by the following recursive process:

1) B is a member of the chain.
2) If X is a member of the chain, then all sets implied (definition 6) by X are added to the chain.

As an example of how a chain is formed, consider Table V. Fig. 1 illustrates the formation of the chain

TABLE V

	I_1	I_2	I_3
1	1, 0	3, 1	4, 1
2	–, 0	5, 1	6, 1
3	–	1, 0	5, 1
4	2, 1	3, 0	4, 0
5	6, 1	–	4, 0
6	3, 1	6, –	1, 0

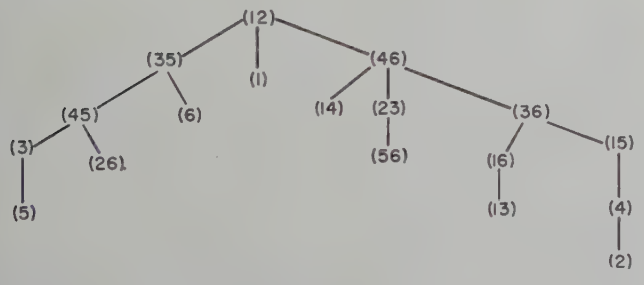


Fig. 1—A chain.

generated by (12). For example, in column I_2, the next-state entries for rows 1 and 2 are 3 and 5, respectively. Hence, the line from (12) to (35) in the figure. Since, in column I_3, the next-state entries for rows 3 and 5 are 5 and 4, respectively, (45) is added to the chain. The process is continued until no new members can be found. The compatibles can now be found with the aid of Theorem 3.

Theorem 3: A set B is a compatible if, and only if, there is no set X in the chain generated by B, such that, for some k, $Z(x_1, I_k)$ and $Z(x_2, I_k)$ are defined and unequal, where x_1 and x_2 belong to X.

When using this theorem to test a given set for compatibility, it is advisable to check the outputs of each set for agreement as the chain is being constructed, since a contradiction early in the process may make it unnecessary to complete the chain. For instance, in the example of Fig. 1, it would have been possible to terminate the test after (35) was shown to belong to the chain because the outputs of rows 3 and 5 are contradictory for input I_3.

Since tests for pairwise compatibility are basic to all of our procedures, it will be useful to have a systematic form for carrying out such tests and recording the results. The *implication table* illustrated in Table VI(b), leads to a particularly convenient and orderly procedure for deriving compatibles. We shall demonstrate the method by means of an example based on the flow Table VI(a).

TABLE VI

	I_1	I_2	I_3	I_4
1	2, –	3, 0	–	4, –
2	3, 0	5, 0	–	–
3	4, –	6, –	3, –	–
4	5, 1	3, 0	–	1, –
5	–	6, 0	–	–
6	–	–, 1	4, –	2, –

(a)

2	23 35				
3	24 36	34 56			
4	25	X	45 36		
5	36	56	$\surd$	36	
6	X	X	34	X	X
	1	2	3	4	5

(b)

2	X				
3	X	X			
4	X	X	45 36		
5	36	X	$\surd$	36	
6	X	X	34	X	X
	1	2	3	4	5

(c)

There is one cell in Table VI(b) for each pair of rows of the given flow table. The first step in our process is to fill these cells column by column, starting at the top of each column in the following manner.

Process C:

1) If there is a contradictory pair of outputs for a given row-pair in any input column, then an X is placed in the corresponding cell of the implication table and no further attention need be given to that row-pair. This condition occurs for (16), (26), (46), (56), and (24) in our example.
2) If, for row-pair *ab*, the outputs are not contradictory, then enter all row-pairs implied by *ab* in the *ab* cell. (It is never necessary to enter *ab* in the *ab* cell.) Thus, in cell (12) we enter (23) and (35), in cell (13) we enter (24) and (36), etc. In cell (35) there are no pairs to be entered, so a check mark is inserted. Note that in cell (14) we enter only (25), omitting (14). At this point, the table is in the form of Table VI(b).

After the first "pass" through the table has been made, a second inspection is made of all cells corresponding to row-pairs that are entries in the table. In our example we inspect, in sequence, the cells corresponding to (23), (35), (24), (36), (25), (36), etc. If the entry in position *ab* is an X, then an X is inserted in all cells that contain *ab*, and any other entries in such cells may be ignored in all further steps. Thus, in this step we insert an X in position (13) since this cell contains a (24) entry. X's are also inserted in the cells corresponding to (23) and (25) because these cells contain (56) as an entry, and cell (56) contains an X. This process is then repeated. The next time, X's are inserted in position (12), which contains a (23) entry, and (14), which contains a (25) entry. On the next inspection, no new X's are added, and the process is thus terminated. For the sake of clarity, the final implication table is shown as a separate table, namely Table VI(c). (In practice, there is no need to redraw the original table.)

All row-pairs whose entries are not X's are compatible, and those with X's are incompatible. The process described here is fully equivalent to the method of Theorem 3 (applied to 2-member sets) and is sufficiently well defined to be programmed on a digital computer.

Definition 11: A *maximal compatible* (MC) is a set of rows which form a compatible and which is not included in any larger compatible.

If we can find all of the MC's, then we have in effect found all compatibles, since an obvious corollary of Theorem 2 is that every subset of a compatible is also a compatible.

We shall use Table VII to illustrate the method. (The non-X cells in this implication table have not been filled in since their contents are irrelevant for the purpose of finding the MC's.)

TABLE VII

	1	2	3	4	5
2	X				
3					
4	X	X	X		
5		X		X	
6					X

Process D: Start from the rightmost column of the implication table and work leftward constructing a list of compatibles as follows.

1) Write down the pairwise compatibles in the first column that has any. In our example, this is (46).
2) Examine the next column to the left (say column k). If k is pairwise compatible with both members of any previously listed 2-member compatibles, then add k to these compatibles, replacing the pairs with 3-member compatibles. Add to the list any other pairwise compatibles in the k column. In our example, for $k=3$ we add (35) and (36) to the list. For $k=2$ we replace (36) by (236).
3) Proceed to the next column on the left which has one or more non-X entries (say f). If f is compatible with all members of some previously listed compatibles, then add f to those compatibles. If f is compatible with all the members of any subsets of previously listed compatibles, then add to the list compatibles consisting of f and these subsets. Also add any other pairwise compatibles in the f column. At each step, eliminate compatibles included in other members of the list. Repeat this process until all columns have been examined. The final list (augmented by any single states not in any pairwise compatibles) will contain all of the MC's. In our example, for $f=1$, we replace (35) with (135) and add (136) to the list. The resulting MC's are therefore (46), (236), (135) and (136).

This is a straightforward process which is not too laborious, and which can be programmed.

There is a dual procedure for finding the MC's which is based on the use of incompatible pairs to break down large sets of rows into smaller and smaller sets until only maximum compatibles remain. The basic operation performed in this process is as follows.

If *abcdef* is being considered as a possible MC, and if *b* and *e* are incompatible, then the candidate set is split into two parts, one with *b* omitted, and the other with *e* omitted, namely *acdef* and *abcdf*. The latter two sets are now candidates, and the next incompatible pair is applied to each in a similar manner. At each stage, candidate sets are omitted if they are contained in other candidate sets.

An efficient way of carrying out these steps is described below, again using Table VII as an example.

Process E: Start with column 1 and work to the right.

1) Write down the set of all states except for 1. Then write down a set of states consisting of 1 followed by all states corresponding to non-X rows in column 1. In our example, we would have at this point (23456) and (1356).
2) Move to the next column containing X's (say column j). Examine those sets on the list that contain j. Replace each such set which also includes at least one state corresponding to rows that are X's in column j by two sets—the original set without j, and the original set without any of the states that are incompatible with j, as indicated by the X's in the j column. Repeat this process for each column, and at each stage eliminate sets included in other sets on the list.

In our example, the list of sets progresses as below after the indicated column is inspected.

1) (23456) (1356)
2) (3456) (236) (1356)
3) (456) (236) (1356)
4) (46) (236) (1356)
5) (46) (236) (135) (136).

The final result, as shown in step 5), is the same as that obtained with Process *D*. A method for finding the MC's without using 2-member compatibles is given in the Appendix.

It is possible to devise variations combining elements of both of the methods described above. For example, in Table VI(c) we might have observed at once that 1 is compatible only with 5 and that 2 is not compatible with any other row. Then (15) and (2) could have been listed at once as MC's, and the incompatibles could have been used to reduce (3456) to (345) and (36).

V. Using the Compatibles

The minimization process can now be restated:

Process F:

1) Find the maximal compatibles, using the procedures of Section IV.
2) Find the smallest closed collection of compatibles.
3) Merge each compatible of the chosen collection into a single row of a new reduced table as described in step 4) of Process *A*.

It would be most satisfying to be able to continue the development by presenting a systematic procedure for selecting a minimal closed set of compatibles. Unfortunately, we have been unable to find any such general method (other than enumeration). The remainder of the paper will be concerned with various isolated tools, miscellaneous results (mainly of a negative nature), and a series of nontrivial examples to illustrate some of the ideas and to show that the situation is really not as bad as the preceding portion of this paragraph would seem to indicate. We have applied the principles evolved here to about two dozen flow tables containing as many as fifteen rows, and have always succeeded in finding a minimum solution within about two hours. The use of a digital computer would no doubt extend the size of the tables that could be reduced.

An upper bound on the number of states in any minimized table is the number of MC's, since the set of all MC's is always closed.

A lower bound can be found in terms of *incompatibles*, which are sets of states that are pairwise incompatible. *Maximal incompatibles* are those incompatibles not included in any other incompatibles those which can be found by interchanging X's and non-X's in the final implication table and using either of the procedures described for finding MC's. The number of elements in the largest maximal incompatible is a lower bound on the number of rows in the reduced table. This was pointed out by Ginsburg.[9]

The following result, which is almost self-evident, is sometimes useful and leads to a simple solution in the case of completely specified flow tables.

Theorem 4: If each MC of Q contains a state not contained in any other MC, then the minimum-state flow table P such that $P \geq Q$ can be synthesized from the closed set of all MC's. The number of states of P is equal to the number of MC's.

Since for a completely specified flow table, the premise of the above result is met "supremely," in that the MC's for a completely specified machine are disjoint, one needs only to find the MC's to find a minimum-state covering flow table for this case.

The following is a variation of reduction Process *F*:.

Process G:

1) Find the MC's (and thereby all of the compatibles).
2) Find the chain generated by each compatible (definition 10). Some compatibles generate identical chains, so that the number of chains will often be less than the number of compatibles.
3) Try all combinations of chains. Within each combination, eliminate compatibles that are included in other compatibles. (If a compatible p appears in a minimum closed collection, then q cannot be a member of that collection if q is a subset of p.) The minimum closed collection of compatibles will consist of one of these reduced collections.
4) Merge each of the selected compatibles into one row of a reduced table.

(Note that in the following examples some of the flow tables are not strongly connected.[3] This fact is of no significance, since each of these tables can be converted to strongly connected tables by adding supplementary input columns. This can always be done without causing any other changes in the problems.)

Example 1: Looking at Table VIII(a), it seems obvious that row 2 can be eliminated immediately by merging it with row 1. The resulting table (with the

merged pair labelled 2 is shown as Table VIII(b). Table IX(a) is the implication table for Table VIII(b) and the MC's are simply (2), (4) and (35). All are required to form the necessary closed set, and so the reduced table having rows A, B and C corresponding to (2), (4) and (35), respectively, is shown as Table IX(b). This table meets both the upper and lower bounds for Table VIII(b).

TABLE VIII

	I_1	I_2	I_3
1	3, 0	5, 1	–
2	3, 0	5, –	–
3	2, –	3, 0	1, –
4	2, 0	3, –	5, –
5	–	5, 0	1, –

(a)

	I_1	I_2	I_3
2	3, 0	5, 1	–
3	2, –	3, 0	2, –
4	2, 0	3, –	5, –
5	–	5, 0	2, –

(b)

TABLE IX

	2	3	4
3	X		
4	$23X$	$25X$	
5	X	$\surd$	$25X$ 35

(a)

	I_1	I_2	I_3
A	C, 0	C, 1	–
B	A, 0	C, –	C, –
C	A, –	C, 0	A, –

(b)

Now let us attempt to reduce the original Table [VIII(a)] directly. The implication table is Table X(a). Using Process E for finding the MC's, the following steps result.

1) (2345) (124)
2) no change
3) (245) (235) (124)
4) (24) (25) (235) (124) so the MC's are (124) and (235).

Thus, using both MC's, we obtain the *two*-row Table X(b), where A and B correspond to (124) and (235), respectively. This table meets both the lower and upper bounds for Table VIII(a).

TABLE X

	1	2	3	4
2	$\surd$			
3	X	35		
4	23 35	23 35	$15X$	
5	X	$\surd$	$\surd$	$15X$ 35

(a)

	I_1	I_2	I_3
A	B, 0	B, 1	B, –
B	B, 0	B, 0	A, –

(b)

By making what appeared to be an obvious merger in the original table, we lost an opportunity to reduce the table to two rows instead of three. The reason, in this case, is that Table VIII(b) is more completely specified than is Table VIII(a). Since any change which *relaxes* the specifications is clearly not allowed, we see that the moral of this example is that it is unwise to make preliminary mergers in a given flow table. Such changes are always safe only in the trivial cases where two rows are identical, or where there is a row with *all* outputs and next-states left unspecified.

Example 2:

TABLE XI

	I_1	I_2	I_3	I_4	I_5	I_6	I_7
1	6, 0	1, –	4, –	3, –	–	–	–
2	–, 1	–	–	–	3, –	4, –	5, –
3	3, –	5, –	–	–	6, 0	2, –	–
4	–	–	6, –	5, –	–, 1	–	1, –
5	1, –	–	1, 1	–	2, –	–	3, –
6	–	4, –	–, 0	2, –	–	5, –	–

(a)

	1	2	3	4	5
2	X				
3	15 36	24 36			
4	35 46	15	X		
5	16 14	23 35	13 26	13 16	
6	14 23	45	25 45	25	X

(b)

Table XI(b) is the implication table for Table XI(a). Using Process D for finding the MC's, the steps are:

1) (45) (46)
2) (45) (46) (35) (36)
3) (245) (246) (235) (236)
4) (245) (246) (235) (236) (145) (146) (135) (136).

Step 4) gives the MC's. The upper bound on the number of necessary rows is thus useless since it exceeds 6. The lower bound is 2.

Our first thought now might be to choose (135) and (246) as the numbers of the closed set, since they include all six states. However, a check of the flow table shows that (135) implies (136), which is not included in either (135) or (246). Thus (135) and (246) do not constitute a closed set. Further efforts at choosing a closed set from among the MC's are balked by similar difficulties as a result of the wealth of implications associated with each compatible.

If we construct a chain of implications starting with any of the pairwise compatibles, we find that *all* the pairwise compatibles will be included. In other words, no collection of compatibles is closed unless every pairwise compatible is included in at least one member of the collection.

Furthermore, if we examine the original flow table, we find that (145) implies (146), which implies (235), and so forth, so that all of the MC's except (246) form a circular chain of implications. Thus, if any of the MC's other than (246) appear in a collection, then that collection will not be closed unless it contains the seven MC's in the chain. But this collection would have more members than the number of rows in Table XI(a) and hence would not correspond to a reduction.

If (246) appeared, then the pairs (25), (23), (14), (15), (16), (45), (13), (35), and (36) would also have to appear. Again, this would not constitute a reduction. Finally, if only pairs appeared, all twelve would be necessary.

Hence we conclude that this flow table cannot be reduced, even though the MC's have three members each. Neither bound is met.

Example 3:

TABLE XII

	I_1	I_2
1	–, 0	–
2	–, 0	1, –
3	1, –	–
4	–	–, 0
5	6, 1	4, –
6	–	5, –

(a)

2	√				
3	√	√			
4	√	√	√		
5	*X*	*X*	16	√	
6	√	15*X*	√	√	45
	1	2	3	4	5

(b)

The maximal compatibles are found using Process *E* as follows:

1) (12346) (23456)
2) (1346) (1234) (3456)

The upper bound is 3 and the lower bound is 2. Note that only (1234) contains a 2 and that this is compatible implies no others. Hence there must be a minimum solution containing (1234). Compatibles with 5 and 6 are now required. If we try (56), then it will be necessary to include another compatible with (45), since (56) implies (45). Suppose we use (456) instead of (56). Then the compatibles (1234) (456) are closed and therefore constitute a solution. Note that the 4 in (1234) serves no purpose and so (123) (456) is also a solution. Let *A* and *B* replace (123) and (456), respectively. Then a reduced version of Table XII(a) is Table XIII, which meets the lower bound.

TABLE XIII

	I_1	I_2
A	*A*, 0	*A*, –
B	*B*, 1	*B*, 0

If we had used (1234), then the unspecified output in the I_2 column of Table XIII would have been fixed at a zero. Thus, when there is an opportunity to replace a compatible by a subset of itself it is desirable to do so, since the reduced table will, in general, then have more unspecified entries.

Example 4:

TABLE XIV

	I_1	I_2	I_3	I_4
1	2, 0	1, 0	3, 0	–
2	2, 0	1, 0	7, 0	–
3	5, 0	1, 0	3, 0	–
4	5, 0	4, 0	3, 0	8, 0
5	5, 0	6, 0	–	–
6	2, 0	6, 0	7, 0	9, 1
7	–	4, 0	7, 0	–
8	5, 0	1, 0	7, 0	8, 0
9	5, 0	1, 0	7, 0	9, 1

(a)

2	37							
3	25	25 37						
4	25	25 14 37	14					
5	25 16	16	16	46 *X*				
6	37	16	25 16 37	*X*	25			
7	14 37	14	14	37	46 *X*	46 *X*		
8	25 37	25	37	14 37	16	*X*	14	
9	25 37	25	37	*X*	16	25 16	14	*X*
	1	2	3	4	5	6	7	8

(b)

Using Process *D*:

1) (78) (79)
2) (78) (79) (69)
3) (78) (79) (569) (58)
4) (478) (79) (569) (58)
5) (123478) (12379) (123569) (12358) are the MC's.

The upper bound is 4 and the lower bound is 2.

The MC's (123478) and (123569) include all states. All pairwise compatibles involving 1, 2 or 3 are included in these two compatibles and, since all entries in the implication table contain 1, 2 or 3, (123478) and (123569) are closed with respect to all implied two-number compatibles. No compatibles with more than two members are implied by either (123478) or or (123569), as may be ascertained from a check of the I_2 column of the flow table (this is the only column with more than two rows appearing as next-state entries). Hence, a solution corresponds to (123478) and (123569) as shown in Table

XV, where A replaces the former and B the latter. The lower bound is met. Further analysis would show that (13478) and (12569) also constitute a closed collection, but the resulting flow table is the same. It is *not* possible to obtain *any* reduction of Table XIV(a) without partitioning the rows into overlapping sets. Hence this example serves to illustrate, in a rather spectacular manner, the weakness of the classical merging procedures as applied to incompletely specified flow tables.

TABLE XV

	I_1	I_2	I_3	I_4
A	$B, 0$	$A, 0$	$A, 0$	$A, 0$
B	$B, 0$	$B, 0$	$A, 0$	$B, 1$

Example 5:

Let us start by choosing two disjoint pairwise compatibles that imply no other sets, namely (25) and (36). We now need sets with states 1 and 4. The compatible (14) implies only (25), and so a solution which meets the new lower bound of 3 rows is (14) (25) (36). Table XVII is the reduced version with rows A, B, and C corresponding to the 3 compatibles. Another solution is (12) (34) (56). It is interesting to note that we cannot obtain a minimum solution if we use the largest MC.

TABLE XVII

	1	2	3	4
A	–	$C, 1$	$B, 1$	$B, 1$
B	$B, 0$	$C, 0$	$A, 0$	$A, 1$
C	$C, 0$	$C, 1$	$B, 0$	$C, 1$

TABLE XVI

	1	2	3	4
1	–	3, 1	5, 1	2, 1
2	5, 0	–	–	–
3	6, 0	6, 1	–	–
4	–	–	2, 1	–
5	–	6, 0	1, 0	4, 1
6	3, 0	–	2, 0	3, 1

(a)

	1	2	3	4	5
2	√				
3	36	56			
4	25	√	√		
5	X	√	X	X	
6	X	35X	√	X	12 34

(b)

Process D yields the MC's (1234), (56) (36) (25) via the following steps.

1) (56)
2) (34) (36) (56)
3) (234) (36) (56) (25)
4) (1234) (36) (56) (25).

Note that the largest maximal incompatibles have two members, so the lower bound on the number of rows is 2. The upper bound is 4.

The table can be reduced to two rows only if a closed collection of compatibles can be found such that (ignoring members common to both compatibles) there are 3 members in each, or 2 members in one and 4 in the other, or 5 in one and 1 in the other. Since no MC has 5 members, we immediately eliminate the last-mentioned combination. The only 4-member compatible is (1234), and it implies (25), (56), and (36), so that it cannot be part of a 2-member collection. There are no 3-member compatibles other than those obtainable from (1234), and we cannot get two disjoint 3-member sets from (1234). Hence, at least 3 compatibles are required and the lower bound cannot be met.

VI. Conclusion

The problem considered here is one in which, given a finite, but very large set, the object is to find those members which are optimal in a specified sense. (The set consists of all flow tables with less than n rows that cover a given n-row flow table, and we are looking for the members with the smallest number of rows.) Since, for such problems, it is always possible to describe what might be termed a completely enumerative solution, in which all members of the set are examined, there is a sense in which all problems of this type are trivial. However, when the sets are extremely large, the completely enumerative procedure may be hopelessly impractical. It is then particularly desirable to develop direct procedures which generate the desired results without any enumeration. Between these extremes there may lie many partially enumerative methods, and in these cases it is frequently difficult to decide when a satisfactory procedure has been found. This will usually depend upon the complexity of the particular examples that are of interest.

The minimization of incompletely specified flow tables is an exceedingly difficult problem, and we have

found no approach that suggests a direct procedure. Our results are as follows:

1) an analysis of the problem which we feel sheds considerable light on its general nature, and which provides a framework for partially enumerative procedures;
2) methods for directly generating the maximal compatibles, which seem to play an important role in the problem;
3) the application of the above ideas to a partially enumerative procedure which seems to be practical for minimizing flow tables containing up to about fifteen rows (without using a computer);
4) methods for obtaining useful upper and lower bounds on the number of rows in the minimum-row tables.

Appendix I

Proofs of Theorems

Proof of Theorem 1

Assume P covers Q. If P contains any state p which does not cover any state of Q, then p can be omitted, and the resulting table will still cover Q. Assume therefore that every state p_i in P covers at least one state of Q.

Let C_i be the set of all states of Q that are covered by p_i $(i=1, 2, \cdots, n)$. We shall now show that the collection of sets (C_i) is a closed collection, hence verifying the first part of the theorem. The second requirement for closure (definition 7) is clearly met, since if q_1 and q_2 belong to C_i and if p_i covers both q_1 and q_2, then for each I_k, $Z(p_i, I_k)=Z(q_1, I_k)$ and $Z(p_i, I_k)=Z(q_2, I_k)$, if both $Z(q_1, I_k)$ and $Z(q_2, I_k)$ are defined. Therefore, $Z(q_1, I_k)$ and $Z(q_2, I_k)$ are equal whenever they are both defined.

If q_1 is a member of C_j, then starting Q in q_1 and P in p_j, the application to both P and Q of any input sequence starting with I_k (and applicable to q_1) must result in P and Q giving the same output sequence (wherever the output of Q is defined), since by definition p_j *covers* q_1. After the first input I_k of the input sequence, P will be in some state p_v, and Q (if its next state is defined) will be in state q_2, which must be a member of C_v, because when the remainder of the input sequence (which can be any input sequence applicable to q_2) is applied to both P and Q, the outputs from both will be the same (wherever the output of Q is defined). Hence by the definition of "covers," p_v covers q_2, and by the definition of C_v, q_2 is a member of C_v. In summary then, for all states q_1 of C_j we have that if $N(q_1, I_k)$ is defined, then it is in C_v. Or C_j *implies* a set which is included in C_v. This is the first condition for closure (definition 7), and completes the proof of the first part of the theorem.

For the second part of Theorem 1 we are given the closed collection $(C_1, \cdots, C_j, \cdots, C_n)$. Using the synthesis procedure described in step 4) of Process A, we construct an n-state flow table P. We intend to prove that $P \geq Q$. In order to do this, we must show that for each state q of Q there is at least one state p of P such that p covers q. In particular, we will show that if q is a member of C_j, then p_j will cover q. We do this with an inductive argument.

Suppose that Q and P are in some pair of states q and p respectively, such that q is a member of the C-set covered by p. Then we will say that a *covering condition* exists.

1) If, with the two machines initially in a covering condition, any *length-one* input sequence applicable to Q is applied to both P and Q, then it is evident from step *b*) of the synthesis procedure that the two machines will again be in a covering condition.
2) Hypothesize that if, with P and Q initially in a covering condition, any *length-m* input sequence applicable to Q is applied to both machines, then they will again end up in a covering condition.
3) Now assume that, with the machines initially in a covering condition, any *length-m+1* input sequence applicable to Q is applied to both P and Q. Then, according to our hypothesis 2), after the first m members of the input sequence have been applied, the machines will again be in a covering condition. The last member of the length-$m+1$ input sequence can be considered as a length-one sequence applied with the machines in a covering condition. According to 1), this again leaves the machines in a covering condition. Therefore, given the hypothesis 2), it follows that if the machines are initially in a covering condition, the application to both of any length-$m+1$ sequence applicable to Q will leave them again in a covering condition. Since the hypothesis has been established for $m=1$ [in step 1], it follows that when P and Q are initially in a covering condition, they will remain in a covering condition after any input sequence applicable to Q is applied to both machines.
4) When the two machines P and Q are in any covering condition, the output of P is the same as the output of Q where the latter is specified. Hence, when P and Q are in any covering condition (and there is at least one for each state of Q), the output sequences from P will be the same as the specified portions of the corresponding output sequences from Q when any input sequence applicable to Q is applied to both machines. Therefore, P covers Q and the theorem is proved.

Proof of Theorem 2

First we wish to show that every subset of a compatible is a compatible. Let S be the set consisting of all compatibles of flow table Q. S is obviously closed. Let T be the collection of all subsets of compatibles. Let $S \cup T$ (the union of S and T) be the collection of all sets which are either in S or T. If we can show that $S \cup T$ is closed, we will have reached our objective. Certainly any

set s in S meets the two conditions required for closure. If set t is in T, it will also meet the closure conditions since:

1) if both q_1 and q_2 are members of t, since t is a subset of some s in S, and each s meets the closure condition, then $Z(q_1, I_k) = Z(q_2, I_k)$ (if both are defined);
2) furthermore, since t in T is included in some s in S, any set *implied* by t (say b) is included in a set *implied* by s. Since a set implied by s is included in some s, b must be included in some s, so b is a member of T.

Therefore $S \cup T$ is a closed collection, hence every member of T is a compatible, as was to be proved. In particular, every pair of states included in a compatible is a compatible.

Now we shall show that if every 2-member subset of a is a compatible, then a is a compatible. Let S be the collection of all compatibles in flow table Q. S is closed. Let T be the collection of those sets, all of whose 2-member subsets are in S. Now we wish to show that $S \cup T$, the collection of all sets either in S or T, is closed.

If s is a member of S, it obviously meets the closure condition.

1) If t is a member of T, and q_1 and q_2 are in t, then by construction of T, the set (q_1, q_2) is in S, and therefore $Z(q_1, I_k) = Z(q_2, I_k)$ when both exist. This is condition 2 for closure.
2) If t implies the set b, then b^2 (any 2-member subset of b) is implied by t^2 (a 2-member subset of t); but t^2 is in S, so if b^2 is implied by t^2, b^2 is in S. Therefore, every 2-member subset of b is in S. Therefore, b is in T.

Proof of Theorem 3

If there is no set X in the chain generated by B such that for some k, $Z(x_1, I_k)$ and $Z(x_2, I_k)$ are defined but not equal (x_1 and x_2 being in X), then the chain generated by B with the addition perhaps of a number of single states forms a closed collection. It obviously meets the output condition for closure [condition 2) of definition 7]. Similarly, the "implication" condition for closure [condition 1)] is met, because the recursive process which generates the chain (definition 10) guarantees that if X is in the chain, every set implied by X is also in the chain. Since B is a member of the chain, and the chain is closed, B is a compatible.

On the other hand, if there is some set X in the chain and some k for which $Z(x_1, I_k)$, $Z(x_2, I_k)$ are both defined and unequal, then B is not a compatible. To show this, we define a *backward chain generated by* X.

1) X is in the backward chain.
2) If Y is in the backward chain, then every set which implies Y is added to the backward chain.

This backward chain has the following two properties:

1) If X is not a compatible then no member of the backward chain generated by X is a compatible.

Assume that at some step in the generating process all members of the backward chain are not compatibles. Then all sets added to the backward chain must imply sets which are not compatible. If Y implies Z (Z being a member of the backward chain), then Y is not compatible, since any closed collection which contains Y must also contain a set which includes Z. But since Z is not a compatible, there can be no such closed collection, so Y is not a compatible. Therefore, if at some point in the generating process all the sets thus far incorporated in the backward chain are not compatible, then all the sets of the backward chain are not compatibles. Since X is not a compatible, no member of the backward chain generated by X is a compatible.

2) Furthermore, the backward chain generated by X contains B (X being a member of the chain generated by B).

Assume that B is not in the backward chain. Assume further that we are building the chain generated by B and at some point none of the members of the partially constructed chain are members of the backward chain generated by X. Then any new member Y added to the chain must be implied by a set Z already in the chain. Since Z is not in the backward chain, neither is Y. Since initially the chain consists of B which is not in the backward chain by assumption, no set in the chain generated by B can be in the backward chain generated by X. This is the contradiction we seek since X is in the chain generated by B.

From 1) and 2) above we conclude that B is not a compatible.

Appendix II

Alternate Process for Generating Maximal Compatibles

1) If x_1 and x_2 are two states for which there is a k such that $Z(x_1, I_k)$ and $Z(x_2, I_k)$ are both defined and *unequal*, then the pair (x_1, x_2) are called *output incompatible*. List all pairs of states which are output incompatible.
2) Use these output incompatibles to break down the set of all states into larger and larger collections of smaller and smaller sets in such a way that the final collection of sets contains all the largest sets, none of which contain an *output incompatible* pair. This may be done in many ways, one of which is identical to Process E, assuming that the output incompatibles are listed as X's in an implication table.
3) We are now left with a collection of sets $(C_1, \cdots, C_j, \cdots, C_m)$. If there is a pair of states in C_j, x_1 and x_2, and a k such that $N(x_1, I_k)$ and $N(x_2, I_k)$ do not both appear in the same C_r, then x_1 and x_2

are called *implication incompatible*. List all implication incompatibles.

4) Again break down the sets of the collection $(C_1, \cdots, C_j, \cdots, C_m)$ with the *implication incompatibles*. Assuming these are listed as X's in an implication table, may be done with Process E.
5) Repeat step 3 on the resulting collection. Continue to cycle through steps 3), 4) and 5) until no more implication incompatibles are found. The resultant collection will consist of all maximal compatibles. The proof is fairly obvious in the light of the previous proofs.

ACKNOWLEDGMENT

The authors would like to thank S. Ginsburg of the National Cash Register Company for his comments on our manuscript, and for making available pre-publication copies[8–10] which indicated the existence of the problem. C. Y. Lee of the Bell Telephone Laboratories stimulated the authors' early work on the subject by providing a series of tailor-made counter examples which demolished a number of hypotheses (including some of his own). E. J. McCluskey, Jr. and R. H. Mayne of the Bell Telephone Laboratories supplied some helpful criticisms.

Logical Machine Design II: A Selected Bibliography*

DOUGLAS B. NETHERWOOD†

Summary—**The bibliography which appeared in the June, 1958, issue of these TRANSACTIONS is extended to a total of 777 titles. The original format is retained, but in this supplement the scope of material is restricted to technical publications pertaining to the logical design of machines.**

INTRODUCTION

ALTHOUGH this bibliography is a supplement to an earlier paper,[1] the criteria for the current selections differ from those of the original compilation. In LMD I (the original bibliography), a wide survey was attempted, with a sampling of papers from many areas related in some way to the abstract design of machines. In LMD II (the present supplement), the range of attention has been narrowed and intensified. LMD I was intended as a stimulus to a wide variety of persons, but LMD II is intended as a reference source for specialists who are working in the field of logical design.

SCOPE

The general area covered by this report has been called the Theory of Machines. However, there may scarcely be two persons who can agree precisely on definitions for the theory of machines or the limits of logical machine design. The difficulties of fixed definitions will be avoided by stating that some of the categories considered to be of the greatest relevance to this subject (listed in order of decreasing emphasis) are

1) Sequential machines
2) Algebras and models for automata,
3) Switching-net theory,
4) Combinational studies,
5) Component structures,
6) Boolean minimization,
7) Graphs and trees,
8) Neurological structures and analogies.

Disagreement may again be expected on the meaning of these categories and on their relative ranking, but it is believed that this listing will suffice as a general guide. Some other areas of lesser importance and occasional relevance are

9) Programming and coding,
10) Translation of language,
11) Reliability,
12) Memory techniques.

As in LMD I, accessibility of publications has been considered. An attempt has been made to cite every significant paper published in English language journals of wide circulation, particularly in issues of recent date. Historical interest has not been a factor in making selections; technical content alone has been considered. Some papers which are as yet unpublished have been listed, but these have existed at least in mimeographed form. A limited number of copies of them are available, in most cases, directly from the authors.

RUSSIAN PUBLICATIONS

In recent years, there has been a great increase in the number of Russian language papers on logical design, especially on the subject of switching theory. This note-

* Manuscript received by the PGEC, April 13, 1959.
† Electronic Components Lab., Wright Air Dev. Center, Wright-Patterson AFB, Dayton, Ohio.
[1] D. B. Netherwood, "Logical machine design: a selected bibliography," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-7, pp. 155–178; June, 1958.

worthy phenomenon is reflected in LMD II, but complete coverage of all such papers has not been attempted. Again, emphasis has been placed on recent publications; persons who are interested in the earlier work of a particular individual can often find it by tracing the author's own references. A list of currently obtainable translations of Russian papers, including many of those cited here, is available from[2]

Morris Friedman, Inc.,
67 Reservoir Street,
Needham Heights 94, Mass.

Reviews

A fairly large number of reviews are cited in this supplement, as well as in LMD I. The reader should not neglect the benefits which are available through the use of competent reviews. In addition to indicating the general quality and competence of a book or paper, a review can

1) refer to other related material;
2) list errata;
3) indicate specific deficiencies as well as strong points of the paper;
4) identify the audience addressed;
5) show where the paper fits into the literature;
6) tell where difficulties may be expected in reading, what preliminary work may be studied, which material is new, what the main contributions are;
7) suggest extension of the work and sometimes even outline a possible approach to be used;
8) give a resume which, in turn, may
   a) serve as a brief introduction to the topic,
   b) aid in following the development of the presentation,
   c) indicate whether reading of the full paper is advisable.

At the same time, it may be well to take into account the special interests of the reviewer himself. If he comments on lack of emphasis on a favored subject, this may sometimes be discounted; but if he announces a valid contribution in his field, this must be given full weight.

In order to provide cross-indexing of reviews, the numbering of LMD I and LMD II is continuous. Several reviews in LMD II pertain to references in LMD I. Those readers who wish to amend their copies of LMD I to show the converse of this relationship may add the following numbers after the entries shown (brackets are omitted here):

4–527	104–495	288–490
21–671	108–614	314–545
46–528	190–659	361–746
72–658	221–539	365–709
91–738	254–672	382–660
92–504, 594	286–615	

Errata

In addition to the changes which have been reported previously,[3] the following entries of LMD I should be changed as indicated:

[148] Title should be "Lattice theoretic properties of frontal switching functions."

[301] Change "[79]" to "[80]."

[396] Source should read, "*Automata Studies* [397], pp. 157–165."

[472] Entry should read, "Zeheb, D. and Caywood, W. P. 'A symbolic method for synthesis of 2-terminal switching circuits,' *Commun. and Electronics*, pp. 690–693; January, 1955."

p. 174. Index word should be "Probabilistic."

Acknowledgment

Sincere appreciation is extended to Dr. E. F. Moore, of the Bell Telephone Laboratories, for suggestions as to sources of material and for the loan of a large portion of his personal card files.

Bibliography

[485] Abbas, S. A., and Critchlow, D. L. "Calculation of flux patterns in ferrite multipath structures," 1958 IRE National Convention Record, pt. 4, pp. 263–267.
[486] Abhyankar, S. "Minimal 'sum of products of sums' expressions of Boolean functions," IRE Trans. on Electronic Computers, vol. EC-7, pp. 268–276; December, 1958.
[487] Aiken, H. H. "The future of automatic computing machinery," *Nachrichtentech. Fachber*, pp. 4, 31–35, 221; 1956.
[488] Akers, S. B., Jr. "On a theory of Boolean functions," *Bull. Amer. Math. Soc.*, vol. 63, p. 350; November, 1957. (Abstract.)
[489] Aldrich, R. W., and Holonyak, N., Jr., "Multiterminal *p-n-p-n* switches," Proc. IRE, vol. 46, pp. 1236–1239; June, 1958.
[490] Amdahl, G. M. Review of [288]. IRE Trans. on Electronic Computers, vol. EC-6, p. 137; June, 1957.
[491] Andrews, L. J. "A Technique for Using Memory Cores as Logical Elements," Electronics Div., The National Cash Register Co., Hawthorne, Calif.; undated.
[492] Anonymous. "Seminar on the engineering applications of mathematical logic (excerpts)," *Automation Express*, vol. 1, pp. 3–4; May, 1958. (Transl. from *Avtom. Telem.*, vol. 18, pp. 950–952; October, 1957.)
[493] Arant, G. W. "A time-sequential tabular analysis of flip-flop logical operation," IRE Trans. on Electronic Computers, vol. EC-6, pp. 72–74; June, 1957.
[494] Ashby, W. R. "An Introduction to Cybernetics," John Wiley and Sons, Inc., New York, N. Y.; 1956. Review [712].
[495] Aubert, K. E. Review of [104]. *J. Symbolic Logic*, vol. 21, p. 321; September, 1956.
[496] Aufenkamp, D. D. "Analysis of sequential machines II," IRE Trans. on Electronic Computers, vol. EC-7, pp. 299–306; December, 1958.
[497] Aufenkamp, D. D., and Hohn, F. E. "Analysis of sequential machines," IRE Trans. on Electronic Computers, vol. EC-6, pp. 276–285; December, 1957.
[498] Bartee, T. C. "The Automatic Design of Logical Networks," M.I.T. Lincoln Lab., Lexington, Mass., Tech. Rept. No. 191; December, 1958. (ASTIA 207170.)

[2] *Editor's Note*—Also of interest in this connection is the semimonthly journal *Technical Translations*, available from the Office of Technical Services, Washington 25, D. C., at $12.00 per year.

[3] Netherwood, *op. cit.*, p. 250; September, 1958.

[499] Bauer, W. F. "Computer design from the programmer's viewpoint," *1958 Proc. EJCC*, pp. 46–51.
[500] Beatson, T. J. "Minimization of components in electronic switching circuits," *Commun. & Electronics*, no. 37 (*Trans. AIEE*, pt. I, vol. 77), pp. 283–291; July, 1958.
[501] Bellman, R. "Sequential Machines, Ambiguity, and Dynamic Programming," RAND Corp., Santa Monica, Calif., Rept. No. P-1597; January 20, 1959.
[502] Bellman, R., Holland, J., and Kalaba, R. "On an Application of Dynamic Programming to the Synthesis of Logical Systems," RAND Corp., Santa Monica, Calif., Rept. No. P-1551; November 18, 1958.
[503] Berlin, R. D. "Synthesis of n-valued switching circuits," IRE Trans. on Electronic Computers, vol. EC-7, pp. 52–56; March, 1958.
[504] Black, M. Review of [92]. *J. Symbolic Logic*, vol. 22, pp. 286–289; September, 1957.
[505] Blankenbaker, J. V. "Logically micro-programmed computers," IRE Trans. on Electronic Computers, vol. EC-7, pp. 103–109; June, 1958.
[506] Blokh, A. S. "Synthesis of (p, q) contact networks," *Dokl. Akad. Nauk SSSR*, vol. 111, no. 5, pp. 1017–1019; 1956. (In Russian.) Review [741].
[507] Bobeck, A. H. "New concept in large-size memory arrays—the twistor," *J. Appl. Phys.*, vol. 29, pp. 485–486; March, 1958.
[508] ——. "A new storage element suitable for large-sized memory arrays—the twistor," *Bell Sys. Tech. J.*, vol. 36, pp. 1319–1340; November, 1957.
[509] Booth, A. D. "Some curiosities of binary arithmetic useful in testing binary computers," *Computers and Automation*, vol. 4, pp. 33–42; August, 1955.
[510] Bratton, D. "The theory of machines," Control Instrument Co., (Burroughs Corp.); May 14, 1958.
[511] Buck, D., "Device, circuit and logical element trends," presented at 1958 EJCC.
[512] Burks, A. W. "The Logic of Fixed and Growing Automata," Willow Run Labs., Engrg. Res. Inst., Univ. of Michigan, Ypsilanti, Mich., Rept. No. 2144-231-T; February, 1958.
[513] Burks, A. W., and Wang, H. "The logic of automata," *J. Assoc. Comp. Mach.*, vol. 4, pp. 193–218, April, 1957; pp. 279–297, July, 1957.
[514] Byerly, R. T., Long, R. W., and King, C. W. "Logic for supplying topological methods to electric networks," *Commun. & Electronics*, no. 39, (*Trans. AIEE*, pt. I, vol. 77), pp. 657–667; November, 1958.
[515] Cadden, W., "Sequential Circuit Theory," Ph.D. dissertation, Princeton Univ., Princeton, N. J.; 1956.
[516] Cagle, W. B. Review of [700]. Proc. IRE, vol. 46, p. 796; April, 1958.
[517] Cagle, W. B., and Chen, W. H. "Designing Low-Level High-Speed Semiconductor Logic Circuits," Bell Telephone System, Monograph No. 3062; 1958.
[518] Calderwood, J. H., and Porter, A. "Pattern recognition in the synthesis of complex switching systems," *J. Electronics & Control*, vol. 4, pp. 466–480; May, 1958.
[519] Caldwell, S. H. "Switching Circuits and Logical Design," John Wiley and Sons, Inc., New York, N. Y.; 1958. Review [736].
[520] Campeau, J. O. "The synthesis and analysis of digital systems by Boolean matrices," IRE Trans. on Electronic Computers, vol. EC-6, pp. 231–241; December, 1957.
[521] Carr, J. W., III. "Languages, logic, learning, and computers," *Computers & Automation*, vol. 7, pp. 21–26; April, 1958.
[522] Cederbaum, I. "Matrices all of whose elements and subdeterminants are 1, −1, or 0," *J. Math. Phys.*, vol. 36, pp. 351–361; January, 1958.
[523] Chen, M. C. "A magnetic core parallel adder," IRE Trans. on Electronic Computers, vol. EC-7, pp. 262–264; December, 1958.
[524] Chomsky, N. and Miller, G. A. "Finite state languages," *Information & Control*, vol. 1, pp. 91–112; May, 1958.
[525] Church, A. "Application of Recursive Arithmetic in the Theory of Computers and Automata," session on Advanced Theory of Logical Design of Digital Computers, Univ. of Michigan, Ann Arbor; June, 1958.
[526] ——. "Binary recursive arithmetic," *J. Mathémat. Pures et Appl.*, ser. 9, vol. 35, pp. 39–55; 1957. (In English). Review [705].
[527] ——. Review of [4]. *J. Symbolic Logic*, vol. 18, p. 347; December, 1953.
[528] ——. Review of [46]. *Ibid.*, p. 68; March, 1953.
[529] Coates, C. L. "General topological formulas for linear network functions," IRE Trans. on Circuit Theory, vol. CT-5, pp. 42–54; March, 1958.
[530] Constantine, G., Jr. "A load-sharing matrix switch," *IBM J. Res. & Dev.*, vol. 2, pp. 204–211; July, 1958.
[531] Copi, I. M., Elgot, C., and Wright, J. B. "Realization of events by logical nets," *J. Assoc. Comp. Mach.*, vol. 5, pp. 181–196; April, 1958.
[532] Crane, H. D. "A high-speed logic system using magnetic elements and connecting wire only," Proc. IRE, vol. 47, pp. 63–73; January, 1959.
[533] Culbertson, J. T. "Mathematics and Logic for Digital Devices," D. Van Nostrand Co., Inc., New York, N. Y.; 1958.
[534] Curtis, A. "Non-Disjunctive Decompositions," Bell Lab. Rept. No. 19, pp. II 1–II 49; February, 1958.
[535] Dantzig, G. B. "Discrete-variable extremum problems," *Operations Res.*, vol. 5, pp. 266–277; April, 1957.
[536] Davis, M. "Computability and Unsolvability," McGraw-Hill Book Co., Inc., New York, N. Y.; 1958.
[537] de Leeuw, K., *et al.* "Computability by probabilistic machines," in "Automata Studies," Annals of Math. Studies No. 34, Princeton University Press, Princeton, N. J., pp. 183–212; 1956. Review [579].
[538] Dickinson, W. E., and Walker, R. M. "Reliability improvement by the use of multiple-element switching circuits," *IBM J. Res. & Dev.*, vol. 2, pp. 142–147; April, 1958.
[539] Duda, W. L. Review of [221]. *J. Symbolic Logic*, vol. 23, pp. 59–60; March, 1958.
[540] Dulmage, L., and Halperin, I. "On a theorem of Frobenius-König and J. von Neumann's game of hide and seek," *Trans. Roy. Soc. Can.*, ser. 3, vol. 49, pp. 23–29; June, 1955.
[541] Dulmage, A. L., and Mendelsohn, N. S. "Coverings of bipartite graphs," *Canad. J. Math.*, vol. 10, no. 4, pp. 517–534; 1958.
[542] Dunham, B. and North, J. H. "The Use of Multipurpose Logical Devices," Res. Publications, IBM Res. Lab., Yorktown Heights, N. Y., Res. Rept. No. RC-9.
[543] Dunham, B., *et al.* "The multipurpose bias device. Part II. The efficiency of logical elements," *IBM J. Res. & Dev.*, vol. 3, pp. 46–53; January, 1959.
[544] Dutka, J. "Some graphical approaches to coding problems," *RCA Rev.*, vol. 18, pp. 466–474; December, 1957.
[545] Ehrenfeucht, A. Review of [314]. *J. Symbolic Logic*, vol. 22, pp. 331–334; September, 1957.
[546] ——. Review of [634]. *J. Symbolic Logic*, vol. 21, pp. 319–320; September, 1956.
[547] Eldert, C., *et al.* "Shifting counters," *Commun. & Electronics*, no. 35 (*Trans. AIEE*, pt. I, vol. 77), pp. 70–74; March, 1958.
[548] Elgot, C. C. "The Non-Existence of Certain Finite Automata Design Algorithms," Session on Advanced Theory of Logical Design of Digital Computers, Univ. of Michigan, Ann Arbor; June, 1958.
[549] ——. Review of [642]. *J. Symbolic Logic*, vol. 22, pp. 334–335; September, 1957.
[550] Elgot, C. C., and Wright, J. B. "Quantifier Elimination in a Problem of Logical Design," Engrg. Res. Inst., Univ. of Michigan, Ann Arbor, Rept. 2144-1087-M, ASTIA 143834; August 25, 1957.
[551] Engelbart, D. C. "A New All-Magnetic Logic System Using Simple Cores" *Digest of Tech. Papers, 1959 Solid-State Circuits Conf.*, Moore School of Elec. Engrg., Univ. of Pennsylvania, Philadelphia: February 12–13, 1959.
[552] Epstein, G. "Synthesis of electronic circuits for symmetric functions," IRE Trans. on Electronic Computers, vol. EC-7, pp. 57–60; March, 1958.
[553] Epstein, M. A. "Algebraic decoding for a binary erasure channel," Res. Lab. of Electronics, M.I.T., Cambridge, Mass., Tech. Rept. No. 340; March 14, 1958.
[554] Everett, C. J., and Whaples, G. "Representations of sequences of sets," *Amer. J. Math.*, vol. 71, pp. 287–293; April, 1949.
[555] Fitch, F. B. "Representation of sequential circuits in combinatory logic," *Phil. of Science*, vol. 25, pp. 263–279; October, 1958.
[556] Ford, G. W., *et al.* "Combinatorial problems in the theory of graphs I, II, III," *Proc. Natl. Acad. Sciences (USA)*, vol. 42, pp. 122–128, March, 1956; pp. 203–208, April, 1956; pp. 529–535, August, 1956.
[557] Ford, L. R., Jr., and Fulkerson, D. R. "Network flow and systems of representatives," *Canad. J. Math.*, vol. 10, no. 1, pp. 78–84; 1958.
[558] Foster, C., and Rapoport, A. "The case of the forgetful burglar," *Amer. Math. Monthly*, vol. 65, pp. 71–76; February, 1958.
[559] Frankel, S. P. "On the minimum logical complexity required for a general purpose computer," IRE Trans. on Electronic Computers, vol. EC-7, pp. 282–285; December, 1958.
[560] Friedberg, R. M. "A learning machine: Part I," *IBM J. Res. & Dev.*, vol. 2, pp. 2–13; January, 1958.

[561] Friedman, J. "Some results in Church's restricted recursive arithmetic," *J. Symbolic Logic*, vol. 22, pp. 337–342; December, 1957.
[562] Ganzhorn, K. "Basic magnetic logic circuits in computers," *Elektron. Rundschau*, vol. 11, pp. 229–234; August, 1957. (In English.)
[563] Gardner, M. "Logic Machines and Diagrams," McGraw-Hill Book Co., Inc., New York, N. Y.; 1958.
[564] Gelernter, H. L., and Rochester, N. "Intelligent behavior in problem-solving machines," *IBM J. Res. & Dev.*, vol. 2, pp. 336–345; October, 1958.
[565] Gianola, U. F. "Nondestructive memory employing a domain oriented steel wire," *J. Appl. Phys.*, vol. 29, pp. 849–853; May, 1958.
[566] ——. "Switching in rectangular loop ferrites containing air gaps," *ibid.*, pp. 1122–1124; July, 1958.
[567] Gianola, U. F., and Crowley, T. H. "The laddic—a magnetic device for performing logic," *Bell Sys. Tech. J.*, vol. 38, pp. 45–72; January, 1959.
[568] Gilbert, E. N. "Gray codes and paths on the *n*-cube," *Bell Sys. Tech. J.*, vol. 37, pp. 815–826; May, 1958.
[569] Gillert, H. "Magnetic switching circuits for representation of logical relations," *Nachrichtentech Z.*, vol. 10, pp. 391–402; August, 1957. (In English.)
[570] Gillespie, R. G., and Aufenkamp, D. D. "On the analysis of sequential machines," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-7, pp. 119–122; June, 1958.
[571] Ginsburg, S. "On the Length of the Smallest Uniform Experiment Which Distinguishes the Terminal States of a Machine," Electronics Div., The National Cash Register Co., Hawthorne, Calif.; 1957. Also in *J. Assoc. Comp. Mach.*, vol. 5, pp. 266–280; July, 1958.
[572] ——. "On the Reduction of Superfluous States in a Sequential Machine," Electronics Div., The National Cash Register Co., Hawthorne, Calif.; September, 1958.
[573] ——. "Some Remarks on Abstract Machines I, II," *ibid.*, January 15, 1958; May 15, 1958.
[574] ——. "A technique for the reduction of a given machine to a minimal state machine," *ibid.;* December, 1958. Also, this issue, p. 346.
[575] Glantz, H. T. Review of [611]. IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, p. 223; September, 1957.
[576] Golomb, S., and Welch, L. R. "Nonlinear shift-register sequences," Jet Propulsion Lab., California Inst. of Tech., Pasadena, Memo. No. 20-149; October 25, 1957.
[577] Golomb, S. W., Welch, L. R., and Delbrück, M. "Construction and properties of comma-free codes," *Biol. Medd. Dan. Vid. Selsk.*, Copenhagen, vol. 23, no. 9, pp. 1–34; 1958. (In English.)
[578] Good, I. J. "Could a machine make probability judgments?" *Computers & Automation*, vol. 8, pp. 14–16, January, 1959; pp. 24–26, February, 1959.
[579] Gorn, S. Review of [537]. IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, p. 217; September, 1957.
[580] Gould, R. "The Application of Graph Theory to the Synthesis of Contact Networks," Ph.D. dissertation, Harvard Univ., Cambridge, Mass.; 1957. Also Bell Lab. Rept. No. 18; 1957.
[581] ——. "Graphs and vector spaces," *J. Math. Phys.*, vol. 37, pp. 193–214; October, 1958.
[582] ——. "A note on contact networks for switching functions of four variables," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-7, pp. 196–198; September, 1958.
[583] Grebenshchikov, V. N. "Method of synthesis of multiterminal contact networks," *Soviet Physics-Dokl.*, vol. 3, pp. 229–233; March/April, 1958. (Transl. into English from *Dokl. Akad. Nauk SSSR*, vol. 119, pp. 278–281; March, 1958.)
[584] Green, E. J., and Meinig, F. R. "Rate as response probability in discrimination learning," *Science*, vol. 126, pp. 1339–1340; December, 1957.
[585] Grisamore, N. T., Rotolo, L. S., and Uyehara, G. U. "Logical design using the stroke function," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-7, pp. 181–183; June, 1958.
[586] Halmos, P. R. "The basic concepts of algebraic logic," *Amer. Math. Monthly*, vol. 63, pp. 363–387; June/July, 1956.
[587] Hamming, R. W. Review of [759]. *J. Assoc. Comp. Mach.*, vol. 5, p. 389; October, 1958.
[588] Harary, F. "The number of oriented graphs," *Mich. Math. J.*, vol. 4, pp. 221–224; 1957.
[589] ——. "On the number of dissimilar graphs between a given graph-subgraph pair," *Canad. J. Math.*, vol. 10, no. 4, pp. 513–516; 1958.
[590] ——. "Structural duality," *Behavioral Science*, vol. 2, pp. 255–265; October, 1957.
[591] Harary, F., and Ross, I. C. "The number of complete cycles in a communication network," *J. Soc. Psych.*, vol. 40, pp. 329–332; 1954.
[592] Har'el, A. "Transistor two-terminal switches," *Commun. & Electronics*, no. 31 (*Trans. AIEE*, pt. I, vol. 76), pp. 328–338; July, 1957.
[593] Harris, J. R. "Direct-Coupled Transistor Logic Circuitry," Bell Telephone System, Monograph No. 3066; 1958.
[594] Henkin, L. Review of [92]. *Bull. Amer. Math. Soc.*, vol. 63, pp. 320–323; September, 1957.
[595] Henle, R. A. "The application of transistors to computers," PROC. IRE, vol. 46, pp. 1240–1254; June, 1958.
[596] Herzfeld, F. "Computer fabrication and circuit techniques," *ibid.*, pp. 1965–1966; December, 1958.
[597] Higonnet, R., and Grea, R. "The Logical Design of Electrical Circuits," McGraw-Hill Book Co., Inc., New York, N. Y.; 1958.
[598] Hirschhorn, E. "Simplification of a class of Boolean functions," *J. Assoc. Comp. Mach.*, vol. 5, pp. 67–75; January, 1958.
[599] Hohn, F. E., Seshu, S., and Aufenkamp, D. D. "The theory of nets," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 154–161; September, 1957.
[600] Holland, J. "Preliminary Report: Cycles and Automaton Behavior," session on Advanced Theory of Logical Design of Digital Computers, Univ. of Michigan, Ann Arbor; June, 1958.
[601] Huffman, D. A. "Problems of Sequential Circuit Synthesis," *ibid.;* June, 1958.
[602] Humphrey, W. S., Jr. "Switching Circuits with Computer Applications," McGraw-Hill Book Co., Inc., New York, N. Y.; 1958. Review [609].
[603] Huzino, S. "On some sequential machines and experiments," *Mem. Fac. Sci., Kyusyu Univ.*, ser. A, vol. 12, no. 2, pp. 136–158; 1958. (In English.)
[604] ——. "Reduction theorems on sequential machines," *ibid.*, no. 2, pp. 159–179; 1958.
[605] Irons, H. R. "A transistor-magnetic core binary counter," PROC. IRE, vol. 46, pp. 1967–1968; December, 1958.
[606] Ivanov, V. I. "Cyclic relay networks and analytical relations in them," *Dokl. Akad. Nauk SSSR*, vol. 104, pp. 239–241; 1955. (In Russian.) Review [673].
[607] Jacobson, H. "On models of reproduction," *Amer. Scientist*, vol. 46, pp. 255–284; September, 1958.
[608] John, E. R., and Miller, J. G. "The acquisition and application of information in the problem-solving process," *Behavioral Science*, vol. 2, pp. 291–300; October, 1957.
[609] Karnaugh, M. Review of [602]. PROC. IRE, vol. 47, p. 346; February, 1959.
[610] Ketchledge, R. W. "Logic for a digital servo system," *Bell Sys. Tech. J.*, vol. 38, pp. 1–17; January, 1959.
[611] Kister, J., *et al.* "Experiments in chess," *J. Assoc. Comp. Mach.*, vol. 4, pp. 174–177; April, 1957. Review [575].
[612] Kron, G. "Tearing, tensors and topological models," *Amer. Scientist*, vol. 45, pp. 401–413; December, 1957.
[613] Lane, A. L., and Turczyn, A. "A high speed *n*-pole, *n*-position magnetic core matrix switch," 1958 IRE NATIONAL CONVENTION RECORD, pt. 4, pp. 246–253.
[614] Lee, C. Y. Review of [108]. IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, p. 129; June, 1957.
[615] ——. Review of [286]. *Ibid.*, p. 129; June, 1957.
[616] Lee, C. Y., and Chen, W. H. "Several-valued combinational switching circuits," *Commun. & Electronics*, no. 25 (*Trans. AIEE*, pt. I, vol. 75), pp. 278–283; July, 1956.
[617] Leichner, G. H. "Designing computer circuits with a computer," *J. Assoc. Comp. Mach.*, vol. 4, pp. 143–147; April, 1957.
[618] Leiner, A. L., *et al.* "Organizing a network of computers to meet deadlines," *1957 Proc. EJCC*, pp. 115–128.
[619] Lipp, J. P. "Topology of switching elements vs reliability," IRE TRANS. ON RELIABILITY AND QUALITY CONTROL, vol. RQC-10, pp. 21–33; June, 1957.
[620] Lockhart, N. F. "Logic by ordered flux changes in multipath ferrite cores," 1958 IRE NATIONAL CONVENTION RECORD, pt. 4, pp. 268–278.
[621] Löfgren, L. "Automata of high complexity and methods of increasing their reliability by redundancy," *Information & Control*, vol. 1, pp. 127–147; May, 1958.
[622] Lorens, C. S. "Theory and applications of flow graphs," Res. Lab. of Electronics, M.I.T., Cambridge, Mass., Tech. Rept. No. 317; July 15, 1956.
[623] Lowenschuss, O. "Non-binary switching theory," 1958 IRE NATIONAL CONVENTION RECORD, pt. 4, pp. 305–317.
[624] Luce, R. D. "Networks satisfying minimality conditions," *Amer. J. Math.*, vol. 75, pp. 825–838; October, 1953.
[625] ——. "A note on Boolean matrix theory," *Proc. Amer. Math. Soc.*, vol. 3, pp. 382–388; 1952. Review [674].
[626] Lunc (Lunts), A. G. "Algebraic methods of analysis and synthesis of contact networks," *Izvest. Akad. Nauk SSSR*, vol. 16, pp. 405–426; 1952. (In Russian.) Review [675].
[627] ——. "Method of synthesis of a (1, *k*)-terminal network," *Dokl. Akad. Nauk SSSR*, vol. 112, pp. 55–57; 1957. (In Russian.) Review [675].

628] ——. "Synthesis and analysis of relay-contact networks with the aid of characteristic functions," *Dokl. Akad. Nauk SSSR*, vol. 75, pp. 201–204; 1950. (In Russian,) Review [675].
[629] Lupanov, O. B. "On the synthesis of contact networks," *Automation Express*, vol. 1, pp. 7–8; September, 1958. (Transl. from *Dokl. Akad. Nauk SSSR*, vol. 119, pp. 23–26; March, 1958.)
[630] MacKay, D. M. "Mindlike behavior in artefacts," *Brit. J. Phil. Science*, vol. 2, pp. 105–121; August, 1951.
[631] MacKay, R. S., and MacIntyre, R. "Ternary counters," IRE Trans. on Electronic Computers, vol. EC-4, pp. 144–149; December, 1955.
[632] Marcovitz, M. W., and Seif, E. "Analytical design of resistor-coupled transistor logical circuits," IRE Trans. on Electronic Computers, vol. EC-7, pp. 109–119, June, 1958; p. 324, December, 1958.
[633] Marcus, M. P. "The detection and identification of symmetric switching functions with the use of tables of combinations," IRE Trans. on Electronic Computers, vol. EC-5, pp. 237–239; December, 1956.
[634] Markov, A. A. "On the continuity of constructive functions," *Uspekhi Matem. Nauk*, vol. 9, no. 3, pp. 226–230; 1954. (In Russian.) Review [546].
[635] ——. "Theory of Algorithms," *Izdatel'stvo Akad. Nauk SSSR*, Moscow-Leningrad; 1954. Review [710].
[636] Mason, S. J. "About such things as unistors, flow graphs, probability, partial factoring, and matrices," IRE Trans. on Circuit Theory, vol. CT-4, pp. 90–97; September, 1957.
[637] Mayeda, W. "The Application of Mathematical Logic to Network Theory," Elec. Engrg. Res. Lab., Univ. of Illinois, Urbana, Interim Tech. Rept. No. 9, Contract DA-11-022-ORD-1983; April 30, 1958.
[638] McCluskey, E. J., Jr. "Iterative combinational switching networks—general design considerations," IRE Trans. on Electronic Computers, vol. EC-7, pp. 285–291; December, 1958.
[639] McCulloch, W. S. "Biological computers," IRE Trans. on Electronic Computers, vol. EC-6, pp. 190–192; September, 1957.
[640] ——. "Stable, Reliable, and Flexible Nets of Unreliable Formal Neurons," Res. Lab. of Electronics, M.I.T., Cambridge, Mass., Progress Rept. No. 49, pp. 118–129; April 15, 1958.
[641] McNaughton, R. "Logical and combinational problems in computer design," *Computers & Automation*, vol. 6, pp. 30–31; January, 1957.
[642] Mealy, G. H. "A method for synthesizing sequential circuits," *Bell Sys. Tech. J.*, vol. 34, pp. 1045–1079; September, 1955. Also Bell Telephone System, Monograph No. 2458. Review [549].
[643] ——. "Notes on Computer Languages and Computer Design," session on Advanced Theory of Logical Design of Digital Computers, Univ. of Michigan, Ann Arbor; June, 1958.
[644] Meltzer, B., and Brown, I. F. "Digital field computers," *Nature*, vol. 181, pp. 1384–1385; May 17, 1958.
[645] Mendelsohn, N. S., and Dulmage, A. L. "Some generalizations of the problem of distinct representatives," *Canad. J. Math.*, vol. 10, no. 2, pp. 230–241; 1958.
[646] Miehle, W. "Link-length minimization in networks," *Operations Res.*, vol. 6, pp. 232–243; March/April, 1958.
[647] Miller, R. E. "Formal analysis and synthesis of bilateral switching networks," IRE Trans. on Electronic Computers, vol. EC-7, pp. 231–244; September, 1958.
[648] Minsky, M. L. "Heuristic Aspects of the Artificial Intelligence Problem," M.I.T. Lincoln Lab., Lexington, Mass., Group Rept. No. 34-55, pp. I 1–I 24; December, 1956.
[649] Moore, E. F. "Table of four-relay contact networks," Appendix III of [597].
[650] Morris, E. F., and Wohr, T. E. "Automatic implementation of computer logic," *Commun. Assoc. Comp. Mach.*, vol. 1, pp. 14–20; May, 1958.
[651] Moskowitz, F. "The analysis of redundancy networks," *Commun. & Electronics*, no. 39 (*Trans. AIEE*, pt. I, vol. 77), pp. 627–632; November, 1958.
[652] Mostowski, A. Review of [684]. *J. Symbolic Logic*, vol. 22, p. 99; March, 1957.
[653] ——. Review of [756]. *Ibid.*, pp. 76–77; March, 1957.
[654] Muller, D. E. "Asynchronous Switching Theory," session on Advanced Theory of Logical Design of Digital Computers, Univ. of Michigan, Ann Arbor; June, 1958.
[655] Mullin, A. A. "Reliable stochastic sequential switching circuits," *Commun. & Electronics*, no. 39 (*Trans. AIEE*, pt. I, vol. 77), pp. 606–611; November, 1958.
[656] Nakagawa, N. "On evaluation of the graph trees and the driving point admittance," IRE Trans. on Circuit Theory, vol. CT-5, pp. 122–127; June, 1958.
[657] Naslin, P. "Circuits à Relais et Automatismes à Séquences," Dunod, Paris; 1958.
[658] Nelson, R. J. Review of [72]. *J. Symbolic Logic*, vol. 18, p. 348; December, 1953.
[659] ——. Review of [190]. *J. Symbolic Logic*, vol. 20, pp. 69–70; March, 1955.
[660] ——. Review of [382]. *J. Symbolic Logic*, vol. 21, p. 401; December, 1956.
[661] Nerode, A. "Linear automaton transformations," *Proc. Amer. Math. Soc.*, vol. 9, pp. 541–544; August, 1958.
[662] Nesenbergs, M., and Mowery, V. O. "Logic synthesis of some high-speed digital comparators," *Bell Sys. Tech. J.*, vol. 38, pp. 19–44; January, 1959.
[663] Netherwood, D. B. "Matrix Representations of Boolean Forms," presented at 5th Annual Symp. on Computers and Data Processing, Denver Res. Inst., Univ. of Denver, Colo.; July 24, 1958.
[664] Neumann, P. "A Markov Model for Sequential Processes," Bell Labs., Inc., Rept. No. 19, pp. III 1–III 56; February, 1958.
[665] Newell, A., Shaw, J. C., and Simon, H. A. "Chess-playing programs and the problem of complexity," *IBM J. Res. & Dev.*, vol. 2, pp. 320–335; October, 1958.
[666] Newman, M. Review of [703]. *J. Assoc. Comp. Mach.*, vol. 5, p. 389; October, 1958.
[667] Oberschelp, W. "Varianten von Turingmaschinen," *Mathematische Logik und Grundlagenforschung*, vol. 4, nos. 1–2, pp. 53–62; 1958.
[668] Ore, O. "Studies on directed graphs," *Annals Math.*, vol. 63, pp. 383–406, May, 1956; vol. 64, pp. 142–153, July, 1956.
[669] Ostianu, V. M. "On the synthesis of switching circuits with selector switches," *Dokl. Akad. Nauk SSSR*, vol. 103, no. 4, pp. 827–830; 1955.
[670] Parker, W. L., and Bernstein, B. A. "On uniquely solvable Boolean equations," *Univ. Calif. Publications in Math.*, new series, vol. 3, no. 1, pp. 1–29; 1955. Review [768].
[671] Pawlak, Z. Review of [21]. *J. Symbolic Logic*, vol. 21, pp. 103–104; March, 1956.
[672] ——. Review of [254]. *J. Symbolic Logic*, vol. 21, p. 104; March, 1956.
[673] ——. Review of [606]. *J. Symbolic Logic*, vol. 21, p. 333; September, 1956.
[674] ——. Review of [625], *J. Symbolic Logic*, vol. 21, pp. 102–103; March, 1956.
[675] ——. Review of [626], [627], [628]. *J. Symbolic Logic*, vol. 22, p. 378; December, 1957.
[676] ——. Review of [687]. *J. Symbolic Logic*, vol. 21, pp. 331–332; September, 1956.
[677] ——. Review of [685]. *J. Symbolic Logic*, vol. 21, p. 332; September, 1956.
[678] ——. Review of [751]. *J. Symbolic Logic*, vol. 21, pp. 332–333; September, 1956.
[679] Penrose, L. S. "Mechanics of self-reproduction," *Ann. Human Genetics*, vol. 23, pt. 1, pp. 59–72; November, 1958.
[680] Percival, W. S. "The solution of passive electrical networks by means of mathematical trees," *Proc. IEE*, vol. 100, pt. 3, pp. 143–150; May, 1953.
[681] Percus, J. K. "Matrix analysis of oriented graphs with irreducible feedback loops," IRE Trans. on Circuit Theory, vol. CT-2, pp. 117–127; June, 1956.
[682] Phister, M., Jr. "Logical Design of Digital Computers," John Wiley and Sons, Inc., New York, N. Y.; 1958. Reviews [714], [748].
[683] Polya, G. "On picture-writing," *Amer. Math. Monthly*, vol. 63, pp. 689–696; December, 1956.
[684] Povarov, G. N. "Concerning the study of symmetric Boolean functions from the point of view of the theory of relay-contact circuits," *Dokl. Akad. Nauk SSSR*, vol. 104, pp. 183–185; 1955. (In Russian.) Review [652].
[685] ——. "Mathematical theory of the synthesis of $(1, k)$-terminal relay-contact networks," *Dokl. Akad. Nauk SSSR*, vol. 100, pp. 909–912; 1955. (In Russian.) Review [677].
[686] ——. "A method for synthesis of computing and controlling contact circuits," *Avtomatika i Telemekhanika*, vol. 18, pp. 145–162; 1957. (In Russian.) Review [742].
[687] ——. "On the synthesis of n-terminal relay-contact networks," *Dokl. Akad. Nauk SSSR*, vol. 94, pp. 1075–1078; 1954. (In Russian.) Review [676].
[688] Prim, R. C. "Shortest connection networks and some generalizations," *Bell Sys. Tech. J.*, vol. 36, pp. 1389–1401; November, 1957.
[689] Prywes, N. S. "Diodeless magnetic shift registers utilizing transfluxors," IRE Trans. on Electronic Computers, vol. EC-7, pp. 316–324; December, 1958.
[690] Putnam, H. "Decidability and essential undecidability," *J. Symbolic Logic*, vol. 22, pp. 39–54; March, 1957.
[691] Quastler, H. "The complexity of biological computers," IRE Trans. on Electronic Computers, vol. EC-6, pp. 192–194; September, 1957.

[692] Rajchman, J. A., and Crane, H. D. "Current steering in magnetic circuits," *ibid.*, pp. 21–30; March, 1957.
[693] Raney, G. N. "Sequential functions," *J. Assoc. Comp. Mach.*, vol. 5, pp. 177–180; April, 1958.
[694] Rashevsky, N. "Contributions to topological biology—some considerations on the primordial graph and on some possible transformations," *Bull. Math. Biophys.*, vol. 18, pp. 113–128; June, 1956.
[695] Reed, I. S. "Some Mathematical Remarks on the Boolean Machine," M.I.T. Lincoln Lab., Lexington, Mass., Tech. Rept. No. 2; December 19, 1951.
[696] ——. "Symbolic Design Techniques Applied to a Generalized Computer," *ibid.*, Rept. No. 141; January 3, 1957.
[697] ——. "Symbolic synthesis of digital computers," *Proc. of the 1952 Meeting, Assoc. Comp. Mach.*, Richard Rimbach Associates, Pittsburgh, Pa., pp. 90–94.
[698] Rey, T. J. "Digital rate synthesis for frequency measurement and control," M.I.T. Lincoln Lab., Lexington, Mass., Tech. Rept. No. 188, ASTIA 205596; October 30, 1958.
[699] Reza, F. "Some topological considerations in network theory," IRE Trans. on Circuit Theory, vol. CT-5, pp. 30–42; March, 1958.
[700] Richards, R. K. "Digital Computer Components and Circuits," D. Van Nostrand Co., Inc., New York, N. Y.; 1957. Review [516].
[701] ——. "New system design techniques," *1958 Proc. EJCC*, pp. 55–51.
[702] Riguet, J. "Algorithmes de Markov et théorie des machines," *Compt. Rend. Acad. Sci.*, vol. 242, pp. 435–437; 1956. Review [709].
[703] Riordan, J. "An Introduction to Combinatorial Analysis," John Wiley and Sons, New York, N. Y.; 1958. Review [666].
[704] ——. "The numbers of labeled colored and chromatic trees," *Acta Math.*, vol. 97, pp. 211–225; August, 1957. Also, Bell Telephone Monograph 2896; November, 1957.
[705] Robinson, J. Review of [526]. *J. Symbolic Logic*, vol. 23, p. 35; March, 1958.
[706] Rochester, N., *et al.* "Tests on a cell assembly theory of the action of the brain, using a large digital computer," IRE Trans. on Information Theory, vol. IT-2, pp. 80–93; September, 1956.
[707] Rogers, H., Jr. "Certain logical reduction and decision problems," *Annals Math.*, vol. 64, pp. 264–284; September, 1956.
[708] Rose, A. "Many-valued logical machines," *Proc. Cambridge Phil. Soc.*, vol. 54, pt. 3, pp. 307–321; 1958.
[709] Rose, G. F. Review of [365] and [702]. *J. Symbolic Logic*, vol. 23, p. 62; March, 1958.
[710] ——. Review of [635]. *Ibid.*, pp. 77–79; March, 1958.
[711] Rosenblatt, F. "The perceptron," *Psych. Rev.*, vol. 65, pp. 386–408; November, 1958.
[712] Rosenblith, W. A. Review of [494]. IRE Trans. on Electronic Computers, vol. EC-6, p. 307; December, 1957.
[713] Rosenfeld, J. L. "Magnetic core pulse-switching circuits for standard packages," IRE Trans. on Electronic Computers, vol. EC-7, pp. 223–228; September, 1958.
[714] Rosenthal, C. W. Review of [682]. *J. Assoc. Comp. Mach.*, vol. 5, pp. 390–391; October, 1958.
[715] Roth, J. P. "Algebraic topological methods for the synthesis of switching systems. I," *Trans. Amer. Math. Soc.*, vol. 88, pp. 301–326; July, 1958.
[716] Rouche, N. "Some properties of Boolean equations," IRE Trans. on Electronic Computers, vol. EC-7, pp. 291–298; December, 1958.
[717] Rowe, W. D., and Royer, G. H. "Transistor NOR circuit design," *Commun. & Electronics*, no. 31 (*Trans. AIEE*, pt. I, vol. 76), pp. 263–267; July, 1957.
[718] Rumble, W. G., and Warren, C. S. "Coincident current applications of ferrite apertured plates," 1958 WESCON Convention Record, pt. 4, pp. 62–65.
[719] Scheinman, A. H. "A numerical-graphic method for synthesizing switching circuits," *Commun. & Electronics*, no. 34 (*Trans. AIEE*, pt. I, vol. 77), pp. 687–689; January, 1958. Also, Bell Telephone System, Monograph No. 2979.
[720] Schubert, E. J. "Matrix analysis of logical networks," *Commun. & Electronics*, no. 35 (*Trans. AIEE*, pt. I, vol. 77), pp. 10–13; March, 1958.
[721] Schützenberger, M. P. "A tentative classification of goal-seeking behavior," *J. Mental Science*, vol. 100, pp. 97–102; January, 1954.
[722] Scott, N. R., "Mathematical logic and switching networks," in "Analog and Digital Computer Technology," to be published by McGraw-Hill Book Co., Inc., New York, N. Y.
[723] Scriven, M. "The mechanical concept of mind," *Mind*, vol. 62, pp. 230–240; April, 1953.
[724] Semon, W. "Synthesis of series-parallel network switching functions," *Bell Sys. Tech. J.*, vol. 37, pp. 877–898; July, 1958.
[725] Seshu, S. "On electric circuits and switching circuits," IRE Trans. on Circuit Theory, vol. CT-3, pp. 172–178, September, 1956; p. 284, September, 1957.
[726] ——. "Theory of Linear Graphs, with Applications in Electrical Engineering," Dept. of Elec. Engrg., Syracuse Univ., N. Y.; 1958.
[727] Shannon, C. E. "Von Neumann's contributions to automata theory," *Bull. Amer. Math. Soc.*, vol. 64, pt. 2, pp. 123–129; May, 1958.
[728] Shapiro, H. S., and Slotnick, D. L. "On the mathematical theory of error-correcting codes," *IBM J. Res. & Dev.*, vol. 3, pp. 25–34; January, 1959.
[729] Shekel, J. "Sketch of an algebra of switchable networks," Proc. IRE, vol. 41, pp. 913–921; July, 1953.
[730] Simon, H. A. "Models of Man," John Wiley and Sons, Inc., New York, N. Y.; 1957.
[731] Simon, H. A., and Newell, A. "Heuristic problem solving: the next advance in operations research," *Operations Res.*, vol. 6, pp. 1–10; January/February, 1958.
[732] Simon, J. M. "Some aspects of the network analysis of sequence transducers," *J. Franklin Inst.*, vol. 65, pp. 439–450; June, 1958.
[733] Sittler, R. W. "Systems analysis of discrete Markov processes," IRE Trans. on Circuit Theory, vol. CT-3, pp. 257–266; December, 1956.
[734] Solomonoff, R. J. "An inductive inference machine," 1957 IRE National Convention Record, pt. 2, pp. 56–62.
[735] Srinivasan, C. V., and Narasimhan, R. "On the Synthesis of Finite Sequential Machines," Tata Inst. of Fundamental Res., Bombay; (ca. 1958).
[736] Stuart, R. W. Review of [519]. Proc. IRE, vol. 46, p. 1887; November, 1958.
[737] Sutherland, W. R., Mugglin, M. G., and Sutherland, I. "An electro-mechanical model of simple animals" *Computers & Automation*, vol. 7, pp. 6–8, 23–25, 32; February, 1958.
[738] Svenonius, L. Review of [91]. *J. Symbolic Logic*, vol. 23, pp. 71–72; March, 1958.
[739] Svoboda, F. "Automatic synthesis of relay circuits," *Automatika i Telemekhanika*, vol. 18, pp. 240–255; 1957. (In Russian.) Review [744].
[740] ——. "The use of indeterminate two-valued Boolean functions for the synthesis of switching circuits," *Československá Akademie Věd, Laboratoř Matematických Strojů, Stroje na zpracování informací*, vol. 2, pp. 209–224; 1954. Review [743].
[741] Szeptycki, P. Review of [506]. *J. Symbolic Logic*, vol. 22, pp. 333–334; September, 1957.
[742] ——. Review of [686]. *J. Symbolic Logic*, vol. 23, pp. 61–62; March, 1958.
[743] ——. Review of [740]. *J. Symbolic Logic*, vol. 22, pp. 99–100; March, 1957.
[744] ——. Review of [739]. *J. Symbolic Logic*, vol. 23, p. 61; March, 1958.
[745] Taube, M. "Man-machine relationships," *Res. & Engrg.*, vol. 5, pp. 18–22; January/February, 1959.
[746] Tompkins, C. B. Review of [361]. IRE Trans. on Electronic Computers, vol. EC-6, p. 223; September, 1957.
[747] Tompkins, H. E., McNaughton, R., *et al.* "General switching theory," Moore School of Elec. Engrg., Univ. of Pennsylvania, Philadelphia, Rept. No. 59-02; Contract AF 33(616)-5886, WADC; October 15, 1958.
[748] Tracy, R. A. Review of [682]. Proc. IRE, vol. 46, pp. 1323–1324; June, 1958.
[749] Trakhtenbrot, R. A. "On operators realizable in logical nets," *Dokl. Akad. Nauk SSSR*, vol. 112, no. 2, pp. 1005–1007; 1957. (In Russian.)
[750] ——. "The synthesis of logical nets whose operators are describable in terms of one-place predicate calculus," *Dokl. Acad. Nauk SSSR*, vol. 118, no. 4, pp. 646–649; 1958. (In Russian.)
[751] ——. "The synthesis of schemata without repetitions," *Dokl. Akad. Nauk SSSR*, vol. 103, pp. 973–976; 1955. (In Russian.) Review [678].
[752] Trent, H. M. "A note on the enumeration and listing of all possible trees in a connected linear graph," *Proc. Natl. Acad. Sciences (USA)*, vol. 40, pp. 1004–1007; October, 1954.
[753] Tsetlin, M. L. "On combinations and subdivisions of non-primitive networks," *Automation Express*, vol. 1, pp. 11–13; July, 1958. (Transl. from *Dokl. Akad. Nauk SSSR*, vol. 118, pp. 488–491; January, 1958.)
[754] Tsetlin, M. L., and Eidus, G. S. "A matrix method for synthesizing multi-contact relay networks in communication and control" (excerpts), *Automation Express*, vol. 1, pp. 13–16; July, 1958. (Transl. from *Elektrosviaz'*, pp. 41–48; April, 1958.)
[755] Unger, S. H. "A computer oriented toward spatial problems," Proc. IRE, vol. 46, pp. 1744–1750; October, 1958.
[756] Uspenskij, V. A. "On calculable operations," *Dokl. Akad. Nauk SSSR*, vol. 103, pp. 773–776; 1955. (In Russian.) Review [653].

[757] Vetukhnovski, F. Ya. "On the number of undecomposable nets and some of their properties," *Dokl. Akad. Nauk SSSR*, vol. 123, no. 3, pp. 391–394; 1958.
[758] von Hippel, A. "Molecular Engineering," Lab. for Insulation Res., M.I.T., Cambridge, Mass., Tech. Rept. No. 101, ASTIA AD 76993; October, 1955.
[759] von Neumann, J. "The Computer and the Brain," Yale University Press, New Haven, Conn.; 1958. Review [587].
[760] Wallace, R. A. "The maze-solving computer," *Proc. of the 1952 Meeting, Assoc. Comp. Mach.*, Richard Rimbach Associates, Pittsburgh, Pa., pp. 119–125.
[761] Wallmark, J. T., and Marcus, S. M. "Integrated devices using direct-coupled unipolar transistor logic," IRE Trans. on Electronic Computers, vol. EC-8, pp. 98–107; June, 1959.
[762] Walter, W. G. "The Living Brain," W. W. Norton and Co., New York, N. Y.; 1953.
[763] Warfield, J. N. "A note on the reduction of switching functions," IRE Trans. on Electronic Computers, vol. EC-7, pp. 180–181; June, 1958.
[764] ——. "Switching circuits as topological models in discrete probability theory," *ibid.*, pp. 251–252; September, 1958.
[765] Weeg, G. P. "Folding of symmetric functions," *ibid.*, p. 325; December, 1958.
[766] Weinberg, L. "Kirchhoff's 'third and fourth laws,'" IRE Trans. on Circuit Theory, vol. CT-5, pp. 8–30; March, 1958.
[767] ——. "Number of trees in a graph," Proc. IRE, vol. 46, pp. 1954–1955; December, 1958.
[768] Weinberger, A., and Smith, J. L. "The logical design of a one-microsecond parallel adder using one-megacycle circuitry," *1956 Proc. WJCC*, pp. 103–108.
[769] Wernick, W. Review of [670]. *J. Symbolic Logic*, vol. 22, pp. 96–97; March, 1957.
[770] Wilkes, M. V., Renwick, W., and Wheeler, D. J. "The design of the control unit of an electronic digital computer," *Proc. IEE*, vol. 105, pt. B, pp. 121–128, 144–146; March, 1958.
[771] Wolfendale, E., Ed. "The Junction Transistor and its Applications," The Macmillan Co., New York, N. Y.; 1958.
[772] Wolpe, H. "Algorithm for analyzing logical statements to produce a truth function table," *Commun. Assoc. Comp. Mach.*, vol. 1, pp. 4–12; March, 1958.
[773] Yokelson, B. J., Cagle, W. B., and Underwood, M. D. "Semiconductor circuit design philosophy for the central control of an electronic switching system," *Bell Sys. Tech. J.*, vol. 37, pp. 1125–1160; September, 1958.
[774] Young, J. Z. "The organization within nerve cells," *Endeavor* vol. 15, pp. 5–19; January, 1956.
[775] Zadeh, L. A. "What is optimal?" IRE Trans. on Information Theory, vol. IT-4, p. 3; March, 1958.
[776] Zemanek, H. "Die Lösung von Gleichungen in der Schaltalgebra," *Arch. elekt. Übertragung*, vol. 12, pp. 35–44; January, 1958.
[777] Zierler, N. "Several binary-sequence generators," M.I.T Lincoln Lab., Lexington, Mass., Tech. Rept. No. 95; September 12, 1955.

Index

(Numbers refer to listings in the Bibliography. A slant line precedes the first word in a title when some other word is cited. Alphabetization is by principal words, ignoring conjunctions, articles, etc.)

C

D

E

F

G

H

I

J

K

L

M

S

T

U

V

W

1

Operational Analog Simulation of the Vibration of a Beam and a Rectangular Multicellular Structure*

A. BEN CLYMER†

Summary—A feasibility study of the use of an operational analog computer for solution of structural problems was undertaken. A beam problem and a rectangular multicellular structure problem were run to test the method. In this paper, which is a progress report, it is shown that the method is highly competitive with digital computer and passive-element computer methods for solution of any structural problem.

Introduction

Structural Problems

ASSEMBLIES comprised of elastic members are of great industrial and military importance. Among the more common structures, one thinks of airframes, automobiles, ships and submarines, bridges, and machine tools. Successful design of these structures requires accurate analysis of a variety of static and dynamic problems.

In turn, structural analysis requires a faithful mathematical representation of (or electrical analogy) to the structure, in order to permit solution by hand or on a computer.

The present paper is concerned primarily with beams and with multicellular structures; *i.e.*, structures comprised of two plates, the space between them being divided into cells by beams. An example of such a structure is a contemporary aircraft or missile wing, in which the plates are called "skins" and the beams are called "spars" and "ribs." If the spars and ribs are orthogonal, and if the plan form is rectangular, then one has a "rectangular multicellular structure." Such a structure is of little practical importance in itself, but it serves to illustrate some of the difficulties arising in the analysis of complicated structures for solution on computers.

Passive-Element Computer Studies

A number of passive electrical circuits are known which are both mathematically and physically analogous to various types of structural members. By interconnecting the electrical circuits properly, one obtains an electrical analogy or model of a structure. This is often called the "direct analogy" method of structural analysis.

The direct analogy method was developed by McCann and MacNeal,[1] and it has since been exploited by Computer Engineering Associates,[2–6] Pasadena, Calif. Many high-performance airframes built in this country are analyzed on passive-element (direct analogy) computers.

Passive-element computers have been applied successfully to the analysis and design of numerous types of structures such as delta and trapezoidal wings, fuselages, wing-fuselage intersections, empennages, wing-nacelle intersections, wing-pylon-store systems, helicopter blades, concrete dams, etc.

The following types of airframe structural problems have been solved on passive-element computers: static stress analysis, static influence coefficients, normal modes of vibration, flutter, aeroelasticity, and response to gusts and blasts. Analogies are known which would permit solution of an even greater variety of problems.

Digital Computer Studies

A digital computer, given a description of a structure, can be programmed to yield, for the structure, the influence coefficient matrix, the normal modes, aeroelastic characteristics, flutter speed, etc. Dynamic problems are usually treated as eigenvalue problems rather than as systems of coupled differential equations of motion.

Some airframe manufacturers have employed only digital computers for structural studies. Others have used digital computers chiefly to check the designs obtained on a passive-element computer.

The great advantage of a digital computer is accuracy. Its great shortcoming is elasped time, and therefore, cost and delay, even in an open-shop facility. These characteristics of a digital computer tend to limit its application to the final stages of airframe design analysis, when design changes are less radical and frequent; or else to the earliest stages of design, when highly simplified analyses are conducted on systematic families of designs in order to discover a preliminary optimum class of designs for study later in greater detail on a passive-element computer.

* Manuscript received by the PGEC, February 2, 1959; revised manuscript received, June 22, 1959. This paper was presented at the Natl. Simulation Conf., Dallas, Tex., October 23, 1958.

† Dynamics Dev. Group, North American Aviation, Inc., Columbus, Ohio.

[1] G. D. McCann and R. H. MacNeal, "Beam vibration analysis with an electric-analog computer," *J. Appl. Mech.*, vol. 72, pp. 13–26; March, 1950.

[2] R. H. MacNeal, "The solution of elastic plate problems by electric analogies," *J. Appl. Mech.*, vol. 18, pp. 59–67; March, 1951.

[3] S. V. Benscoter and R. H. MacNeal, "Equivalent Plate Theory for a Straight Multicell Wing," NACA Tech. Note 2786; September, 1952.

[4] S. V. Benscoter and R. H. MacNeal, "Analysis of Straight Multicell Wings on Cal-Tech Analog Computer," NACA Tech. Note 3113; January, 1954.

[5] R. H. MacNeal and S. V. Benscoter, "Analysis of Multicell Delta Wings on Cal-Tech Analog Computer," NACA Tech. Note 3114; December, 1953.

[6] R. H. MacNeal and S. V. Benscoter, "Analysis of Sweptback Wings on Cal-Tech Analog Computer," NACA Tech. Note 3115; January, 1954.

Operational Analog Computer Studies

In practically every field of analytical engineering, operational analog computers, in competition with other types of computers, have found a valuable niche by virtue, principally, of the rapidity with which changes can be made in the system as simulated. One would expect then that engineers would have sought to apply operational analog computers to structural design problems. Indeed, many engineers have done so, but unfortunately few have been privileged to conduct much of a feasibility study. Little of value has been published concerning studies of structures more complicated than single beams, except studies in which the structure was represented by its normal modes rather than in detail.

A number of inquiries by the writer indicate that some authors and organizations had undertaken to solve at least one structural problem (generally a four to six cell beam or beam-rod) in detail on an operational analog computer. They are Howe, Howe, and Rauch, University of Michigan, Ann Arbor;[7] S. Paine, Convair, San Diego, Calif.;[8] W. Stark, Convair, Pomona Calif.;[8] Lockheed, Burbank, Calif.; H. Reisman, Martin, Denver, Colo.; M. Johnson, North American Aviation, Downey, Calif.; North American Aviation, Inglewood, Calif.; Lockheed, Marietta, Ga.; P. J. Hermann, Goodyear Aircraft Co., Akron, Ohio; M. Yanowitch, Project Cyclone, New York, N. Y.[9] Results ranged from very poor to very good. None of these organizations is known to have continued after the initial study.

Beam Studies

Error Amplification and Ill-Conditioning

One difficulty, which must have plagued several of the early studies, is "error amplification," *i.e.*, a small change in a resistance value (potentiometer setting or resistor temperature) causing an astonishingly large change in the answers. Error amplification was explained and analyzed mathematically in the case of static deflection of a uniform cantilever beam by R. MacNeal, Computer Engineering Associates, in an unpublished internal report dated August 27, 1958. MacNeal showed that error amplification occurs in operational analog circuits for structures when fourth differences are formed by a single amplifier, which requires that individual elements of the stiffness matrix be mechanized by individual potentiometers. In such a circuit an error in a single potentiometer setting or input resistance corresponds directly to an error in a single element of the stiffness matrix. Now it is widely recognized that most practical structures have stiffness matrices which are "ill-conditioned," which means that "the determinant of the . . . matrix is small, in magnitude, relative to certain of the cofactors of elements of that matrix."[10] Therefore, MacNeal argued, one would expect serious error amplification in an operational analog structural simulation in which the stiffness matrix is mechanized explicitly.

One of the purposes of the studies reported herein was to verify MacNeal's theoretical prediction. It was decided to start with a simple beam problem, in which causes of any difficulties could be tracked down. A cantilever beam was represented in terms of six lumped masses. Rotary inertia and shear flexibility were not included.

Two different mechanizations were employed: 1) with fourth differences formed in single amplifiers, potentiometers being used to represent individual elements of the stiffness matrix, and 2) with four successive first differences formed in four successive amplifiers, potentiometers being used to represent mass and stiffness. The first mechanization would show error amplification, it was conjectured, but the second mechanization would not.

Study of Stiffness Matrix Mechanization

The equation of motion of a uniform beam may be written in the form:

$$\rho\ddot{w} = -M'',$$

where

$$M = EIw'',$$

in which ρ is the mass per unit length, w the deflection at station x and time t, dots denote differentiation with respect to time, primes denote differentiation with respect to span-wise distance, M is the bending moment at station x and time t, and EI is the bending stiffness. Replacing the primes by finite difference approximations, one obtains

$$\ddot{w}_i = -\frac{1}{\rho(\Delta x)^2}(M_{i+1} - 2M_i + M_{i-1}),$$

and

$$M_i = \frac{EI_i}{(\Delta x)^2}(w_{i+1} - 2w_i + w_{i-1}),$$

where i is the station number, EI_i is the section stiffness at station i, and Δx is the distance between adjacent stations. Let $i=1$ denote the free tip point. Then the equations of motion of the lumped masses may be displayed in the following form:

[7] C. E. Howe, R. M. Howe, and L. L. Rauch, Engrg. Res. Inst., Univ. of Michigan, Ann Arbor, Memo. UMN-67; January 1951. See also, R. M. Howe and V. S. Haneman, Jr., "The solution of partial differential equations by difference methods using the electronic differential analyzer," Proc. IRE, vol. 41, pp. 1497–1508; October 1953.

[8] "Simulation council newsletter," *Instr. and Automation*, vol. 30, pp. 1515–1520; August, 1957.

[9] M. Yanowitch, "Solution of a Flutter Problem," Project Cyclone Rept.; June, 1955.

[10] F. B. Hildebrand, "Introduction to Numerical Analysis," McGraw-Hill Book Co., Inc., New York, N. Y, p. 439; 1956.

$$\begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{w}_1 \\ \ddot{w}_2 \\ \ddot{w}_3 \\ \ddot{w}_4 \\ \ddot{w}_5 \\ \ddot{w}_6 \end{bmatrix}$$

$$= -\frac{EI}{\rho(\Delta x)^4} \begin{bmatrix} +1 & -2 & +1 & 0 & 0 & 0 \\ -2 & +5 & -4 & +1 & 0 & 0 \\ +1 & -4 & +6 & -4 & +1 & 0 \\ 0 & +1 & -4 & +6 & -4 & +1 \\ 0 & 0 & +1 & -4 & +6 & -4 \\ 0 & 0 & 0 & +1 & -4 & +7 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{bmatrix}.$$

The diagonal matrix on the left is the mass matrix, and the symmetric square matrix on the right is the stiffness matrix.

The above matrix equation was solved on an IBM 704 computer for the first three modes as a check on the operational analog results. The results of a digital solution are compared with analog results in Table I.

TABLE I

Quantity	First Mode		Second Mode		Third Mode	
	Analog	Digital	Analog	Digital	Analog	Digital
w_1	1.000	1.000	1.000	1.000	1.000	1.000
w_2	0.766	0.773	0.197	0.190	−0.358	−0.354
w_3	0.530	0.550	−0.460	−0.470	−0.774	−0.767
w_4	0.329	0.343	−0.768	−0.773	0.028	0.036
w_5	0.154	0.169	−0.659	−0.652	0.880	0.886
w_6	0.048	0.048	−0.281	−0.274	0.688	0.686
Freq. (cps)	0.0444	0.0443	0.255	0.254	0.638	0.637

All potentiometers were set to the nearest 0.01 volt (out of 100 volts) by means of a 4-digit digital voltmeter, which is standard practice on operational analog computers. It may be concluded that, with no special pains, one can obtain the first three modal frequencies for a 6-mass beam within 0.5 per cent error, using the stiffness matrix mechanization on an operational analog computer. However, the mode shape errors can be as large as 2 per cent of full scale.

It will be noted that the largest errors are in the first mode shape. This is presumably due to the time scale, which was chosen to make the third mode frequency less than 1 cps in order to avoid dynamic errors, but which made the net input voltage to the force integrator in the first mode small enough to be close to the noise level. Considerably better results were obtained by running with larger integrator input gains (faster).

Excellent results were also obtained in a case of uniform static loading, as is shown in Table II.

The foregoing runs having adequately confirmed the accuracy of the operational analog computer, even as mechanized, when using normal care in setting potentiometers, it became of interest to know the effect upon static and dynamic results due to an error, for example, of 0.5 per cent in various matrix elements. In particular, it was of interest to verify MacNeal's error analysis.

Table III illustrates the effect upon the calculated static deflection of a cantilever beam under uniform load of a ± 0.5 per cent error in setting the value of K_{33}, the element of the stiffness matrix in the third row and third column (nominally 6).

TABLE II

Deflection	Run 12A Analog	Run 12B Analog	(Converted) Digital
	(volts)	(volts)	(volts)
w_1	29.17	29.24	29.33
w_2	22.78	22.82	22.90
w_3	16.49	16.52	16.56
w_4	10.54	10.56	10.57
w_5	5.37	5.39	5.373
w_6	1.59	1.59	1.585

TABLE III

Deflection	Error in K_{33}		
	−0.5 per cent	0	+0.5 per cent
	(volts)	(volts)	(volts)
w_1	84.12	29.24	17.95
w_2	66.16	22.82	13.92
w_3	48.32	16.52	9.98
w_4	30.82	10.56	6.39
w_5	15.54	5.39	3.29
w_6	4.50	1.59	0.99

The foregoing findings confirm the great sensitivity of the cantilever-beam equations to a small change in a single element of the stiffness matrix. A factor-of-three error in deflection at the tip due to an error of only 0.5 per cent in K_{33} is surely astonishing, especially in view of the reasonably small size of the deflection errors in the runs reported previously.

Consider the physical significance of an error in K_{33} or any other diagonal element of the stiffness matrix for a structure. Such an error can be simulated as an undesired spring to ground. If the element is increased, the spring is of the ordinary positive type, running from mass number 3 to ground; if the element is decreased, the spring is negative. Of the two, a negative spring is much more damaging to accuracy of results since it is not physically realizable as a passive element and therefore acts as an energy source.

A negative spring to ground pushes a mass farther away from zero deflection, and the farther it deflects, the harder it pushes. This runaway effect stops only when the portion of the structure between the affected mass and the support of the structure is sufficiently bent to counteract the force from the negative spring. It is to be noted in Table III that a positive spring to ground has much less effect.

An error in an off-diagonal element of a stiffness matrix can be simulated by means of a device which

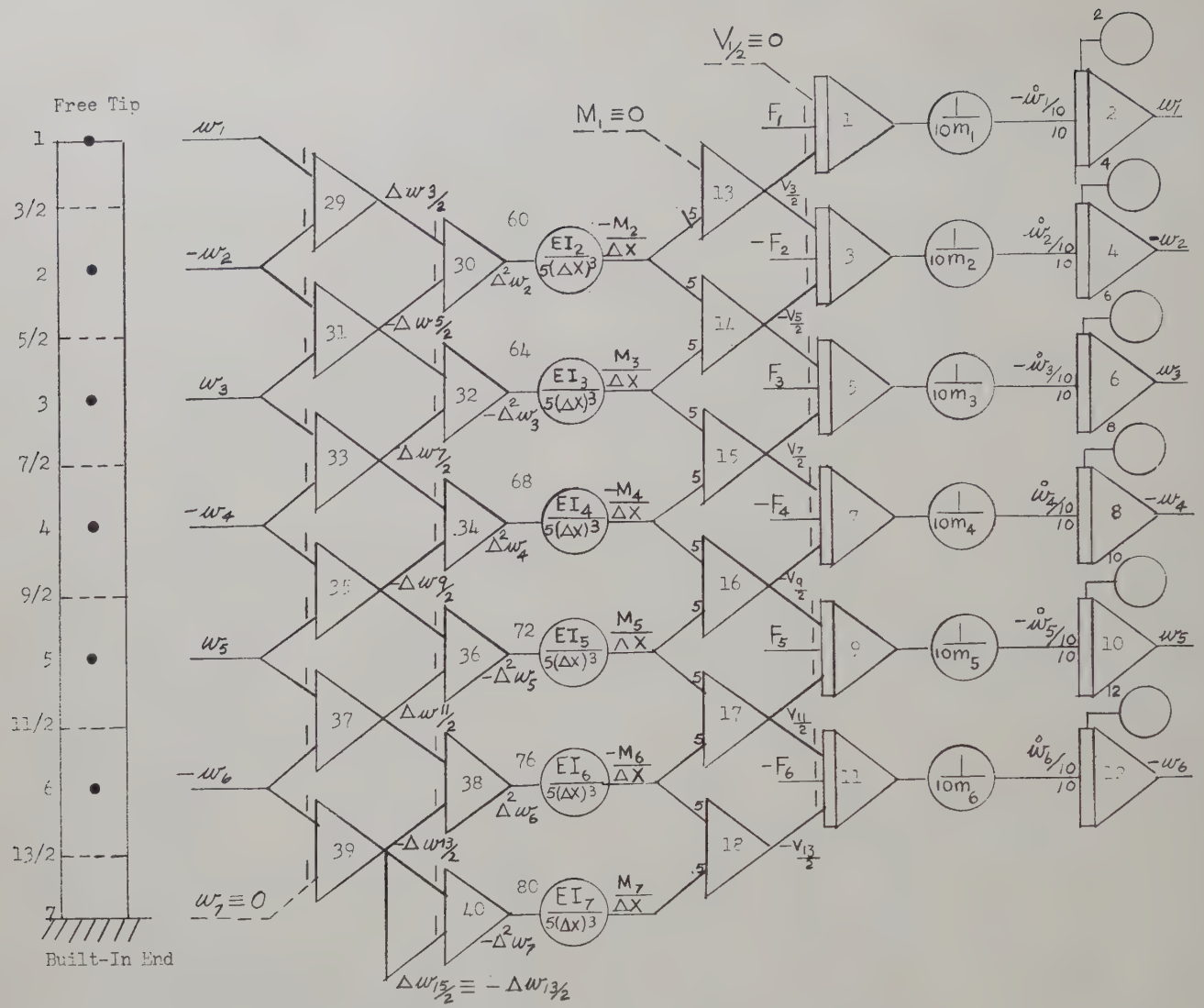


Fig. 1—Operational analog mechanization for a 6-cell beam.

measures the absolute deflection of one mass point and applies a proportional force to another mass. This, too, is an active element.

Similar error amplification occurs in dynamic studies.

As one would expect, a certain percentage error in every element of one row of the stiffness matrix was found to produce an unamplified error in structural behavior since the row error is equivalent to a lumped mass element or lumped stiffness element error.

It is concluded that the computer mechanization employing the formation of a fourth difference in a single amplifier is unsatisfactory because of the large amount of error amplification involved.

Study of First-Difference Mechanization

Next, the same beam was studied using the mechanization shown in Fig. 1. A feature of this mechanization is that all quantities of interest are available as outputs. The equations used were as follows:

$$\Delta w_{i+1/2} = w_{i+1} - w_i$$

$$\Delta^2 w_i = \Delta w_{i+1/2} - \Delta w_{i-1/2}$$

$$M_i = -\frac{EI_i}{(\Delta X)^2}\Delta^2 w_i$$

$$V_{i+1/2} = \frac{1}{\Delta X}(M_{i+1} - M_i)$$

$$m_i \ddot{w}_i = F_i + V_{i+1/2} - V_{i-1/2}$$

where w_i is the deflection station at i, M_i is bending moment, EI_i is flexural rigidity, ΔX is the station spacing, V is shear force, m_i is the ith lumped mass, and F_i is the applied force. The values chosen were: $m_1 = 0.1$, all other m's $= 0.2$, $EI_i/(\Delta X)^3 = 1.703$.

The foregoing equations are precisely those underlying the direct analogy method on a passive-element computer.

The study was remarkably successful, as shown by the tabulation of the worst errors obtained in Table

TABLE IV

Mode	Frequency	Mode Shape
1	0.04 per cent	0.07 per cent
2	0.1 per cent	1 per cent
3	0.1 per cent	Not obtained
Static	—	0.5 per cent

TABLE V

First Mode Results

Item	Analog Run Number					Digital
	2	6	8	14	20*	
f_1 (cps)	0.044298	0.044299	0.044297	0.044299	0.044294	0.44325
w_1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
w_2	0.7725	0.7726	0.7722	0.7725	0.7720	0.7726
w_3	0.5499	0.5495	0.5497	0.5494	0.5500	0.5498
w_4	0.3432	0.3428	0.3430	0.3430	0.3428	0.3431
w_5	0.1689	0.1690	0.1685	0.1688	0.1691	0.1692
w_6	0.0474	0.0478	0.0480	0.0479	0.0477	0.0477

* Run $\sqrt{10}$ times faster than with scaling shown in Fig. 1.

TABLE VI

Second Mode Results

Item	Analog Run Number			Digital
	8	14	20*	
f_2 (cps)	0.25478	0.25462	0.25472	0.25445
w_1	−1.0000	−1.0000	−1.0000	−1.0000
w_2	−0.1856	−0.1873	−0.1979	−0.1900
w_3	0.4813	0.4744	0.4564	0.4699
w_4	0.7807	0.7786	0.7528	0.7726
w_5	0.6522	0.6564	0.6347	0.6520
w_6	0.2715	0.2759	0.2642	0.2739

* Run $\sqrt{10}$ times faster than with scaling shown in Fig. 1.

TABLE VII

Third Mode Results

Item	Analog Run Number				Digital
	8	14	20*	21**	
f_3 (cps)	0.6375	0.63755	0.63693	0.6372	0.63690

* Run $\sqrt{10}$ times faster than with scaling shown in Fig. 1.
** Run $\sqrt{10}$ times slower than with scaling shown in Fig. 1.

IV, and the complete results displayed in Table V through Table X.

No error amplification was observed. In fact, the largest percentage error in an output quantity was only 70 per cent as large as the input percentage error deliberately introduced. Therefore, it is concluded that error amplification is not inherent in operational analog solution of a structural problem. Error amplification would seem to occur only when the mechanization involves the formation of higher order differences in a single amplifier.

TABLE VIII

Static Deflection Under Uniform Load

Item	Analog Run Number			Digital
	10	15	16	
(volts)				
w_1	97.23	97.26	97.25	97.77
w_2	76.01	75.95	75.94	76.34
w_3	54.97	54.92	54.92	55.20
w_4	35.09	35.05	35.05	35.23
w_5	17.84	17.82	17.82	17.91
w_6	5.26	5.26	5.26	5.29
(normalized)				
w_1	1.0000	1.0000	1.0000	1.0000
w_2	0.7809	0.7809	0.7809	0.7808
w_3	0.5647	0.5647	0.5647	0.5646
w_4	0.3605	0.3604	0.3604	0.3604
w_5	0.1833	0.1832	0.1832	0.1832
w_6	0.0540	0.0541	0.0541	0.0541

TABLE IX

Effects of Gain Errors on First Mode Frequency

Gain Error Imposed	f_1 (cps)	Per Cent Error Re Digital
1 per cent in w_1 into $\Delta w_{3/2}$	0.044368	0.10 per cent
No error; op. analog	0.044298	−0.05 per cent
No error; digital	0.044325	0
1 per cent in w_2 into $\Delta w_{3/2}$	0.044242	−0.20 per cent
1 per cent in w_2 into $\Delta w_{5/2}$	0.044439	0.25 per cent
1 per cent in w_6 into $\Delta w_{13/2}$	0.044313	−0.025 per cent

TABLE X

Effects of Gain Errors on Static Tip Deflection

Gain Error Imposed	w_1 (volts)	Per Cent Error re Digital	Per Cent Error re Analog
None; digital	97.77	0	—
None; analog	97.34	−0.4 per cent	0
1 per cent in w_4 into $\Delta w_{7/2}$	97.64	−0.1 per cent	0.3 per cent
1 per cent in $\Delta w_{7/2}$ into $\Delta 2w_4$	96.66	−1.1 per cent	−0.7 per cent
5 per cent $V_{7/2}$ into $m_4\ddot{w}_4$	98.92	1.2 per cent	1.6 per cent

The largest damping ratio observed was of the order of 0.0001, which is 100 times smaller than that which is inescapable in a passive element computer.

All vibration studies were made by using digital mode shapes as deflection initial conditions. The same results could have been obtained, with somewhat more effort, by seeking resonance with a low-frequency oscillator, establishing resonance, removing the excitation, throwing the computer into "Hold," and reading out the mode shape on the digital voltmeter. All frequencies were determined by measuring 10 periods with a counter.

Rectangular Multicellular Structure Studies

Study of Stiffness Matrix Mechanization

As a second test of the use of an operational analog computer for the solution of structural problems, a rec-

tangular multicellular tructure problem was studied. The structure is pictured in Fig. 2, which is taken from Benscoter and MacNeal[4] *i.e.*, a report of a passive-element computer study of the structure. Equations of motion and boundary conditions for the structure are given in the Appendix.

Table XI indicates the agreement obtained for symmetric bending frequencies. All methods agreed on the shape of the first mode (first span-wise bending) within 3 per cent of full scale.

The operational analog and digital mode shapes for the second mode (a saddle) agreed within 7 per cent of full scale. However, this mode was not reported by Benscoter and MacNeal. It appears to have been overlooked, since such a mode was obtained in another rectangular multicellular structure [shown in their Fig. 48(b)] and in a square cantilever plate.[2]

The operational analog and digital mode shapes for the third mode (first chord-wise bending) agreed within 3 per cent of full scale. Neither set of results agreed well with the passive-element computer results, although there was a resemblance.

The operational analog and digital mode shapes for the fourth mode (second span-wise bending) differed by as much as 15 per cent of full scale because of error amplification. Even the digital results by two methods show slight differences in the fourth significant digit.

There are two explanations for the discrepancies with the passive-element computer results:

	Operational Analog	*Passive-Element*
Lumped masses located:	at nodal points	at midpoints
Beam shear:	neglected	included

These simplifications were made in the operational analog computer in the interest of expediency.

As in the first beam study, it was found that large error amplification occurred in the operational analog computer when a stiffness matrix mechanization was used.

Study of First-Difference Mechanization

Next, the same problem was solved on an operational analog computer, forming first differences in individual amplifiers, with and without beam shear. (This work was done after the oral presentation of this paper in Dallas. It will be reported in full in a later paper. Only the highlights will be offered here.) No error amplification occurred. In fact, the largest static errors were about 0.15 per cent, compared with digital results. Mode shapes and frequencies were in much better agreement with digital results than they had been in the first study. For example, the frequency error in the fifth mode was only 2 cycles out of 550.

When beam shear was included, significant discrepancies with passive-element computer results were still present. The saddle mode was still obtained. These discrepancies are ascribed chiefly to the difference in mass lumping, together with some passive-element computer error.

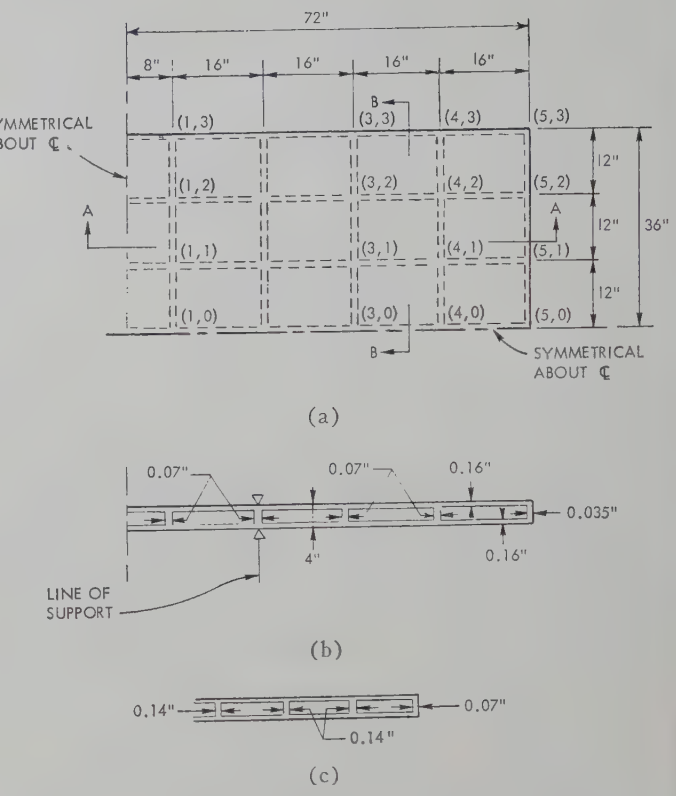


Fig. 2—Rectangular multicellular structure; (a) plan form, (b) section A-A, (c) section B-B.

TABLE XI

SYMMETRIC BENDING FREQUENCIES

Mode	Digital Computer* Program 1	Digital Computer* Program 2	Operational Analog	Passive-Element**
1	47.513 cps	47.513 cps	48.4 cps	48.0 cps
2	265.32	265.32	264.5	Not reported
3	307.94	307.94	303.8	261
4	384.22	384.22	382.4	339

* Program 1 was the conventional matrix iteration method; program 2 sought the eigenvalues by an automatic trial-and-error process in solving the characteristic equation.

** From Benscoter and MacNeal.[4]

Other Possible Mechanizations

Howe, *et al.*,[7] employed a mechanization in which two second differences were formed in two successive amplifiers. The published results indicate that the error amplification in this mechanization was not serious for the small structures studied, but one would expect difficulty with more complicated structures. Moreover, this mechanization does not yield shear forces as outputs, unless additional amplifiers are used.

Larrowe[11] and others at the University of Michigan devised a direct analogy scheme for mechanizing each lumped mass and spring in a mechanical system by means of a combination of an integrator and a potenti-

[11] V. L. Larrowe, "Direct simulation bypasses mathematics, simplifies analysis," *Control Engrg.*, vol. 1, p. 25 ff.; November, 1954.

ometer. Although this scheme often leads to superfluous amplifiers, the circuit obtained can be reduced to the same circuit one would get from first-difference equations.

Extensions of the Method

Aerodynamic Forces

Consider an unswept rectangular wing. From aerodynamic strip theory one can determine the force to apply to any nodal point, given the instantaneous deflection rate and slope of each stream-wise strip of structure, which in turn can be obtained from the nodal point deflections and rates of deflection. Then, for example, using steady-state aerodynamics one has a means for solving the static aeroelastic problem, in which the elastic deformation of the structure interacts with the aerodynamic forces. Furthermore, by employing unsteady aerodynamic theory, one can obtain the dynamic interactions of structural deformation and aerodynamic force, which permits determination of flutter speed and the damping of each normal mode in simulated flight.

At constant Mach number and altitude it is quite simple to find unsteady aerodynamic forces on a strip of structure. The term which depends upon the vertical acceleration of the structure can be neglected or lumped into the structural mass. The rest of the aerodynamic force is directly proportional to the local "effective" angle of attack, which in general lags behind the actual angle of attack. This lag can be simulated with precision by a Duhamel integral of the Wagner function. The minimum-amplifier mechanization for an operational analog computer requires four amplifiers per strip. Aerodynamic influence coefficients, rather than strip theory, would require more equipment. Treatment of the aerodynamic forces due to varying camber would also be more complicated but entirely possible.

Automatic determination of flutter speed could be achieved in principle by making speed variable in the computer (which requires multipliers and function generators) and by use of a rectifying-smoothing-differentiating-smoothing circuit for obtaining a flutter speed error signal to vary the speed automatically in the proper direction. One would start at a speed above the flutter speed, in order to excite only the flutter mode. If the response of the circuit were rapid and smooth, one could slowly vary the altitude and thus be able to plot the flutter boundary automatically in the altitude-speed plane. It would seem that this could be done in less than a minute of running time, which would be in striking contrast to the elapsed time involved on either a passive-element or digital computer.

Gusts and blasts give rise to aerodynamic forces which involve not only the deformation of the structure but also an air motion within the atmosphere. These too could be simulated.

Equations for all of the foregoing effects are well known in the aircraft industry.

Structural Complications

Few structures are as simple as those for which equations are herein derived. Yet one would wish to be able to simulate any structure on an operational analog computer. Therefore, one must devise ways to deal with at least the following types of structural complications.

1) Pylon-tank, pylon-bomb, landing gear, and nacelle-engine appendages on a structure are important in design analysis. These complications require additional equations of motion coupled with the equations of motion of the basic structure. Sometimes the unsteady aerodynamic forces on such appendages are critical.
2) Carry-through structure, by means of which a lifting surface is fastened to a fuselage, is usually quite a complicated type of structure in three dimensions.
3) A multiplicity of closely-spaced spars and ribs would require too many equations of motion to permit assignment of one equation to each spar-rib intersection. It would be necessary to develop a technique requiring fewer equations.
4) Lack of an x-y plane of symmetry in the structure introduces major complications in the equations of motion. For example, the two skins require separate treatment. Beams suffer dilation as well as bending. Cutouts pose an especially interesting situation.
5) Control surfaces fastened to nonrigid actuators introduce complications in the boundary conditions at the root. In a dynamic simulation, the actuator output impedance as a function of frequency can be an important structural parameter.
6) A fuselage has equations of motion which are different in form from those for a multicellular structure. One would have to develop almost as much theory and as large a bag of tricks for fuselages as for wings.
7) Skin panels can buckle under compression and/or shear loads. This affects the stiffness and hence behavior of the structure as a whole. Panels which can buckle in the middle of a computer run would require a circuit for determining at all times whether buckling could occur, and would need a relay amplifier to make the necessary changes in parameters when buckling does occur. Some problems to be run on a structures computer would be concerned entirely with buckling.
8) Elementary beam, plate, and shell theory is based on the assumption that displacements are small compared with thickness. If deflections are larger, the equations become much more complicated in form, but they could be handled in principle.
9) Yielding, plastic flow, creep, etc., require major modification of the equations.

10) The spring constants of rivets in their holes are nonlinear functions. Free play in rivets, fold hinges, and control surface hinges introduce further nonlinearities, which make the vibration frequencies strongly a function of amplitude.

Techniques for handling most of the foregoing and other structural complications are at present being made available within several organizations.

Internal Damping

In addition to inertia and elastic terms, composite structures have damping terms in their equations of motion due to internal damping of various types. It would be highly advantageous to simulate internal damping to an engineering degree of accuracy, one reason being that the flutter speed of a structure depends appreciably upon internal damping.

Usually the most important type of internal damping in a built-up structure is that due to dry friction when the members comprising the structure slide on each other. It should be possible to accomplish general derivations yielding the forms of the forces at each nodal point due to internal stick-slip, and it should be possible to determine experimentally the friction coefficients involved. In a computer one would need, at each nodal point, a relay amplifier to compare the local interface shear friction with the local shear differential tending to cause slip. The relay amplifier would apply forces to surrounding nodal points accordingly. In this relaxation process the inertia would be negligible and the changes in internal load could be considered to be instantaneous.

Another type of internal damping, which is often a cause of design problems, is due to the sloshing of fuel. Under some conditions, the fuel can introduce impulses to the structure in such a phase as to act like negative damping. Therefore it would be desirable to be able to simulate fuel slosh in all directions, at all amplitudes, and over the relevant band of frequencies.

Solid damping, such as that of a tuning fork in a vacuum, is believed to be negligible in aircraft structures.

Thermal Effects

Operational analog computers have long been used successfully for the solution of unsteady heat flow problems. Thus it is natural to wish to incorporate into a structures computer the ability to determine local temperatures as functions of time, in order to find not only the resulting changes in the local modulus of elasticity but also the thermal stresses and structural deformations due to nonuniform heating. Moreover, in principle, one could solve the aerodynamic heating equations at representative surface points, taking into account the actual deformations and motions of the structure and thus closing the thermal-structural loop. Such problems are becoming more important in contemporary airframe design.

If structural damping due to stick-slip has been mechanized, then it would be easy to incorporate the effect of interface pressure upon heat flow across the interface, such as from a skin into a spar, even though this effect is not crucial to the heat flow problem.

Structural Optimization

In principle, therefore, an operational analog computer could solve the structural, aerodynamic, and thermal equations, as well as rigid body and control system equations, simultaneously. This possibility would permit rapid structural optimization by testing each design variation under all required conditions on the computer, and then modifying the structure as is found to be necessary or desirable. Moreover, since all structural parameters would appear in fairly simple potentiometer formulas, structural design changes could be made conveniently on such a computer. Thus, an operational analog structures computer should serve admirably as the backbone of structural design effort.

Evaluation of Method

Problem Setup

At the time of oral presentation of this paper, lack of a simple and rapid procedure for setting up a complicated structural problem on paper, and then on an operational analog computer, appeared to be a formidable obstacle to practical application of the method. There was need for a manual of standard circuits corresponding to a wide variety of types of structural members so that to set up a problem diagram one would merely interconnect the circuits corresponding to the given structural members. Now a fairly complete manual has been prepared, in draft form, at the Columbus Division of North American Aviation, Inc.

Accuracy

It is interesting to note that the transformer parasitic shunt resistance and shunt inductance cause error amplification in a passive-element computer, although by careful selection of scale factors the errors may be kept within reasonable limits (3 per cent) for beams with as many as ten cells.

Thus the operational analog computer (even if mechanized in terms of the stiffness matrix) and the passive-element computer are not equivalent in their sensitivity to stiffness matrix element errors inherent in the computers. In most practical problems the errors in a passive-element computer due to parasitic effects would probably outweigh any errors in an operational analog computer. Hence an operational analog computer would be the better tool for these problems.

On the whole, then, it would appear that an operational analog computer of contemporary design, capable of having its potentiometers set to four significant digits in conjunction with a digital voltmeter (a standard accessory), is more accurate than a passive-element computer.

Reliability

Passive elements of good quality are inherently more reliable than active elements like amplifiers. Hence one would suppose that a passive-element computer is more reliable than an operational analog computer.

It should be kept in mind, however, that so-called "passive-element" computers, to an appreciable extent, consist of amplifiers, current generators, electronic multipliers, electronic function generators, power supplies, etc., requiring 10 to 20 kva. It is these components which set the reliability of a passive-element computer.

Furthermore, in view of findings at Project Cyclone and at other experienced operational analog facilities, one can conclude that there is no known limit imposed on problem size by lack of computer reliability.[12] Moreover, Project Cyclone's findings were based on a computer built in 1952, which is a long time ago in the operational analog computer market. Therefore, lack of reliability is really not a problem with either type of computer in a well-maintained facility.

Conclusions

In the feasibility study reported herein, no technical obstacle preventing successful solution of any practical structural problem on an operational analog computer has been found. The only major obstacle found is a temporary one; namely, the lack of a published manual permitting easy and rapid setup of a problem on paper and on a computer. Once this obstacle is overcome, it should be possible for users of operational analog computers to add the solution of structural problems to their normal repertoire.

The operational analog method shows good promise of capturing an important role in the solution of structural design problems.

Appendix

General Equations for a Rectangular Multicellular Structure

Benscoter and MacNeal[3] have derived general equations for a rectangular multicellular structure. These may also be derived from Lagrange's equations. One can write them in the following form:

$$\rho\ddot{w} = q + \frac{\partial^2 M^x}{\partial x^2} - 2\frac{\partial^2 M^{xy}}{\partial x \partial y} + \frac{\partial^2 M^y}{\partial y^2}, \tag{1}$$

$$M^x = -D^x\left(\frac{\partial^2 w}{\partial x^2} + \nu r_S \frac{\partial^2 w}{\partial y^2}\right), \tag{2}$$

$$M^y = -D^y\left(\frac{\partial^2 w}{\partial y^2} + \nu r_R \frac{\partial^2 w}{\partial x^2}\right), \tag{3}$$

$$M^{xy} = (1-\nu)D^{xy}\frac{\partial^2 w}{\partial x \partial y}; \tag{4}$$

where

ρ = mass of structure per unit area in (x, y) plane,
w = deflection at (x, y),
q = external load per unit area,

$$D^x = \frac{E}{l^y}(I^{Px} + I^S),$$

$$D^y = \frac{E}{l^x}(I^{Py} + I^R).$$

$$D^{xy} = EI^P,$$
$$I^{Px} = I^P l^y,$$
$$I^{Py} = I^P l^x,$$

$$I^P = \frac{h(l^z - h)^2}{2(1-\nu^2)},$$

h = skin thickness,
l^z = total depth of structure,
E = Young's modulus,
ν = Poisson's ratio,
I^S = area moment of inertia of spar section,
I^R = area moment of inertia of rib section,

$$r_S = \frac{1}{1 + I^S/I^{Px}}$$

$$r_R = \frac{1}{1 + I^R/I^{Py}},$$

l^x = distance between ribs,
l^y = distance between spars.

When (1)–(4) are written in finite difference form, they are suitable for solution on an operational analog computer:

$$\frac{m_{i,j}\ddot{w}_{i,j}}{l^x l^y} = \frac{1}{(\Delta x)^2}(M^x_{i+1,j} - 2M^x_{i,j} + M^x_{i-1,j}) + \frac{1}{(\Delta y)^2}(M^y_{i,j+1} - 2M^y_{i,j} + M^y_{i,j-1}) - \frac{2}{\Delta x \Delta y}(M^{xy}_{i+1/2,j+1/2} + M^{xy}_{i-1/2,j-1/2} - M^{xy}_{i+1/2,j-1/2} - M^{xy}_{i-1/2,j+1/2}), \tag{5}$$

$$M^x_{i,j} = -D^x_{i,j}[(w_{i+1,j} - 2w_{i,j} + w_{i-1,j}) + \nu r_S(w_{i,j+1} - 2w_{i,j} + w_{i,j-1})], \tag{6}$$

$$M^y_{i,j} = -D^y_{i,j}[(w_{i,j+1} - 2w_{i,j} + w_{i,j-1}) + \nu r_R(w_{i+1,j} - 2w_{i,j} + w_{i-1,j})], \tag{7}$$

$$M^{xy}_{i+1/2,j+1/2} = (1-\nu)D^{xy}_{i+1/2,j+1/2}(w_{i+1,j+1} + w_{i,j} - w_{i+1,j} - w_{i,j+1}). \tag{8}$$

Shear forces in the structure can be found, if desired in some members, from the finite difference form of the

[12] A. Karen and B. D. Loveman, "Large problem solutions at project cyclone," *Instr. and Automation*, vol. 29, pp. 78–83; January, 1956.

following well-known expressions:[13]

$$Q^x = \frac{\partial M^x}{\partial x} - \frac{\partial M^{xy}}{\partial y} \tag{9}$$

$$Q^y = \frac{\partial M^y}{\partial y} - \frac{\partial M^{xy}}{\partial x}. \tag{10}$$

If one wishes to solve a vibration problem involving a rectangular structure, by means of formulating it as an eigenvalue problem on a digital computer, w in (5) can be replaced by $-\omega^2 w$, and the moments can be replaced by (6)–(8). By combining all coefficients of each $w_{i,j}$ on the right-hand side, one obtains the stiffness matrix.

Boundary conditions at a free edge, as shown by MacNeal,[2-4] involve three conditions: 1) bending moment normal to the edge is zero at the edge, 2) twisting moment one-half cell outboard of the edge is zero, and 3) shear force one-half cell outboard of the edge is zero. These conditions are exactly equivalent to the finite difference form of the Kirchhoff boundary conditions[13] applied at the free edge. Williams and Birschall[14] present the conditions to be employed at a free corner; they agree with NacNeal.[2] Boundary conditions for simply supported edges, clamped edges, elastically supported edges, etc., are offered by MacNeal[2] and Timoshenko.[13]

Plate Equations

Timoshenko[13] derives the following equations for a plate, neglecting rotary inertia, vertical shear, and all other forms of strain energy:

$$\frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y} = -q + \rho \ddot{w},$$

$$Q_y = \frac{\partial M_y}{\partial y} - \frac{\partial M_{xy}}{\partial x}$$

$$Q_x = \frac{\partial M_x}{\partial x} - \frac{\partial M_{xy}}{\partial y}$$

$$M_x = -D\left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2}\right)$$

$$M_y = -D\left(\frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2}\right)$$

$$M_{xy} = (1+\nu) D \frac{\partial^2 w}{\partial x \partial y};$$

where Q is shear force per unit length, q is distributed applied pressure, ρ is mass per unit plan view area, w is deflection normal to the plate surface, M_x is bending moment causing curvature along the x direction, M_{xy} is twisting moment, D is flexural rigidity, and ν is Poisson's ratio.

The corresponding equations in finite difference form for the (i, j) nodal point are:

$$l^y Q^x_{i+1/2,j} - l^y Q^x_{i-1/2,j} + l^x Q^y_{i,j+1/2} - l^x Q^y_{i,j-1/2} + F_{i,j} = m_{i,j} \ddot{w}_{i,j}$$

$$l^x Q^y_{i,j-1/2} = \frac{l^x}{l^y} M^y_{i,j+1} - \frac{l^x}{l^y} M^y_{i,j} + M^{xy}_{i-1/2,j+1/2} - M^{xy}_{i+1/2,j+1/2}$$

$$l^y Q^x_{i+1/2,j} = \frac{l^y}{l^x} M^x_{i+1,j} - \frac{l^y}{l^x} M^x_{i,j} + M^{xy}_{i+1/2,j-1/2} - M^{xy}_{i+1/2,j+1/2},$$

$$\frac{l^y}{l^x} M_{i,j} = -\frac{l^y}{(l^x)^2} D_{i,j}\left[w_{i+1,j} - 2w_{i,j} + w_{i-1,j} + \nu\left(\frac{l^x}{l^y}\right)^2 (w_{i,j+1} - 2w_{i,j} + w_{i,j-1})\right]$$

$$\frac{l^x}{l^y} M^y_{i,j} = -\frac{l^x}{(l^y)^2} D_{i,j}\left[w_{i,j+1} - 2w_{i,j} + w_{i,j-1} + \nu\left(\frac{l^y}{l^x}\right)^2 (w_{i+1,j} - 2w_{i,j} + w_{i-1,j})\right]$$

$$M^{xy}_{i+1/2,j+1/2} = \frac{(1-\nu) D_{i+1/2,j+1/2}}{l^x l^y} (w_{i+1,j+1} - w_{i+1,j} - w_{i,j+1} + w_{i,j}).$$

General Structural Equations

In general, a given structure can be broken down into beam segments, plate modules, shell modules, torsion rod segments, etc. Such a unit of structure will usually have one chief degree of freedom which can be measured by a generalized coordinate. One would hope to be able to neglect kinetic energy in all other degrees of freedom. These others then have equations of equilibrium instead of equations of motion.

In this way one obtains, hopefully, only one equation of motion plus one or more equations of equilibrium. These equations determine the generalized coordinate in terms of internal shear forces, bending moments, twisting moments, tensile forces, etc. These internal loads can then in turn be expresssed in terms of the generalized coordinates by means of Hooke's law, etc.

This general technique should apply to the majority of practical structural design problems. It should yield equations which adapt themselves to a mechanization which is free of error amplification.

[13] S. Timoshenko, "Theory of Plates and Shells," McGraw-Hill Book Co., Inc., New York, N. Y., ch. 4; 1940.

[14] D. Williams and P. C. Birschall, "Solution of the Problem of the Cantilever Square Plate by the Use of Influence Coefficients and a High Speed Digital Computer," Royal Aircraft Estab., Farnborough, Hauts, Eng., Tech. Note Structures 178; 1956.

ACKNOWLEDGMENT

Associated with the author in most activities reported herein were J. Haas and C. R. Walli, of North American Aviation, Inc., Columbus Division. The analysis presented in the Appendix was conducted by J. Haas, based on an analysis of boundary conditions by C. R. Walli. An earlier derivation of an equation of motion for an interior point of a rectangular multicellular structure, not included herein, was written by the author in collaboration with J. Stocker, of North American Aviation, Inc., Columbus Division. A general analysis of error amplification not given herein, is due to J. Haas; it took as its point of departure a uniform cantilever beam analysis by R. MacNeal, of Computer Engineering Associates, Pasadena, Calif. J. Haas and R. Gongaware conducted the operational analog computer studies. J. Stocker, C. R. Walli, J. J. Nichol, J. Italiano (all of North American Aviation, Inc., Columbus Division), and W. Brunner, of Electronic Associates, Inc., Princeton Computation Center, participated in one or more of the operational analog studies of the multicellular structure. M. Iimura conducted all of the digital computer studies.

The Design of Position and Velocity Servos for Multiplying and Function Generation*

EDWARD O. GILBERT†

Summary—**The design of position and velocity servos used in analog computation and simulation for multiplying and function generation is considered. The important characteristics of potentiometers, gear train, motor, amplifier, and tachometer are defined and discussed. Nonlinear performance requirement, such as velocity and acceleration limits, overshoot for large step inputs, and static resolution, are defined in terms of component parameters. A minimum gear reduction ratio is determined on the basis of acceleration, frictional torque ratio, overshoot for large step inputs, or static resolution. Linear system analysis is made and related to system components and nonlinear performance; in particular, it is shown that static resolution is limited by servo amplifier bandwidth for given motor, potentiometers, and gear train. The selection of damping methods and the reduction of steady-state errors is described. An example design is considered.**

INTRODUCTION

THE use of servo positioned potentiometers for multiplication and function generation in analog computation and simulation is widespread. Typically, the potentiometers are driven by a two-phase servomotor through a gear train. The motor, in turn, is driven by an ac amplifier that receives a compensated error signal in modulated form from a synchronous chopper or other modulated waveform source. The performance of such a system is specified by such factors as bandwidth, static resolution, and velocity and acceleration limits.

* Manuscript received by the PGEC, March 12, 1959; revised manuscript received, June 22, 1959.

† Dept. of Aeronaut. and Astronaut. Engrg., University of Michigan, Ann Arbor, Mich.

The design problem is to select suitable components, such as motor, gear train, and potentiometers, so that performance specifications may be realized or optimized. The particular difficulty of design is that these performance requirements are interrelated, and it is not generally possible to specify arbitrarily all the performance characteristics. It is the purpose of this paper to express the interrelationships so that compatible requirements may be understood and a successful design accomplished.

COMPONENTS

There are a number of component characteristics that are significant in design. Many of these characteristics are primarily determined by factors such as accuracy, reliability, availability, economy, and the state of the art. For example, a high resolution, high linearity requirement would specify multiple turn, wire-wound potentiometers. This, in turn, would set quite definite values for the potentiometer inertia and frictional torque. In other cases, less stringent requirements would allow a more flexible selection of components, and hence, a wider range of component characteristics. Individual components will now be discussed.

Potentiometers

Important characteristics for servo design are:

T_p = total potentiometer frictional torque,
I_p = total potentiometer inertia including potentiometer coupling,

η = resolution of the reference potentiometer,
= number of potentiometer wires in full scale,
n_p = number of shaft revolutions in full scale.

For film potentiometers, η is essentially infinite. Desirable characteristics are low T_p and high η, which have rather definite limits set by potentiometer size, resistance, accuracy and n_p. As will be seen, I_p is not usually significant in design.

Gear Train

Important gear train characteristics are:

n_g = gear reduction ratio between motor shaft and reference potentiometer,
I_g = gear inertia referred to the motor shaft,
T_g = maximum gear train frictional torque referred to the motor shaft,
ζ = gear backlash at the potentiometer shaft expressed as a fraction of full-scale potentiometer travel.

The gear reduction ratio is an important parameter fixed by design. The inertia I_g is a function of n_g but may be considered independent of n_g over a limited range which is sufficient for design considerations. The gear train precision is specified by T_g and ζ. Gear friction is a rather erratic function of gear position, and T_g should be small relative to potentiometer and tachometer friction referred to the motor shaft in order to insure jerk-free operation. The backlash ζ should be considerably less than δ, the static resolution error expressed as a fraction of full scale. δ is discussed more fully at a later point. When some of the potentiometers are driven through gearing from other potentiometers, the reference potentiometer should be the one most directly geared to the motor, minimizing backlash in the closed loop. The gear backlash between all potentiometers should be considerably less than $1/\eta$, so that wire-to-wire oscillation of the reference potentiometer is not possible without the frictional torque load of the other potentiometers.

Tachometer

A tachometer may be used for damping or as the velocity reference in a velocity servo. The tachometer is coupled directly to the motor shaft to eliminate gear backlash in the velocity feedback loop. Important characteristics are:

T_t = tachometer frictional torque,
I_t = tachometer inertia including coupling,
K_t = voltage constant in volts/rpm.

Amplifier

Important characteristics are:

K = linear region static gain in volts/full scale error,
e_s = saturation voltage of amplifier,
$Y_a(s)$ = equivalent transfer function of amplifier in linear range, where s is the Laplace transform variable.

The required amplifier gain K will be determined by design. The amplifier should have sufficient power capability so that the saturation voltage e_s approximately equals the maximum motor voltage e_{max}.

The existence and determination of the transfer function $Y_a(s)$ is open to question. This is apparent from the ac amplifier gain $|\overline{Y}_a(j\omega)|$ shown in Fig. 1. Here ω_0 is the carrier frequency and $\omega_0+\omega_m$ and $\omega_0-\omega_m$ are the sideband frequencies corresponding to a sinusoidal modulation frequency of ω_m. For Y_a to exist, $\overline{Y}_a(\pm j\omega_0 \pm j\omega) = Y(\pm j\omega)$. This is certainly not true for $\omega_m \geq 1/\tau_a$, due to the nonsymmetrical gain characteristic about the frequency ω_0. The exact effect of the nonsymmetric gain, including the demodulation characteristics of the motor, would be a worthy subject of analysis.

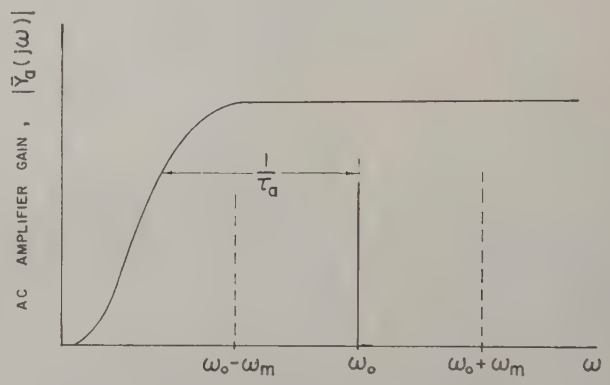


Fig. 1—Typical ac amplifier gain characteristic.

For the design problem considered here, the equivalent transfer function

$$Y_a(s) = \frac{K}{\tau_a s + 1} \tag{1}$$

has proven adequate. $1/\tau_a$ is taken as the lower break frequency of the ac amplifier. Obviously, $1/\tau_a < \omega_0$, and as $1/\tau_a$ approaches ω_0, the transfer function given by (1) becomes less valid, and additional time lags should be considered.

In some servos, a magnetic amplifier provides modulation and amplification. Typically, additional gain is provided by a dc amplifier. The above transfer function is still valid where τ_a is now the time constant of the control winding. Magnetic amplifiers commonly suffer two disadvantages: first, the bandwidth $1/\tau_a$ is considerably less than ω_0; and second, the dynamic characteristics are poor in the saturated state. The bandwidth may frequently be improved using current feedback in the control winding. The poor saturated amplifier dynamics are characterized by long recovery times from the saturated state, which means that the full linear band-

width is not usable. Such poor dynamics are minimized by good amplifier design or by introducing saturation at a point before the magnetic amplifier.

Two-Phase Servomotor

The important servomotor characteristics are obtained from the speed-torque curves for the motor. A typical family of such curves is shown in Fig. 2 in dimensionless form, where T is the motor torque, $\dot{\theta}_m$ is the motor angular velocity in rad/sec, and e is the rms voltage applied to the variable phase. $\dot{\theta}_{m\ \max}$ and $T_{\max}$ are the maximum velocity and torque when $e = e_{\max}$, the maximum motor voltage.

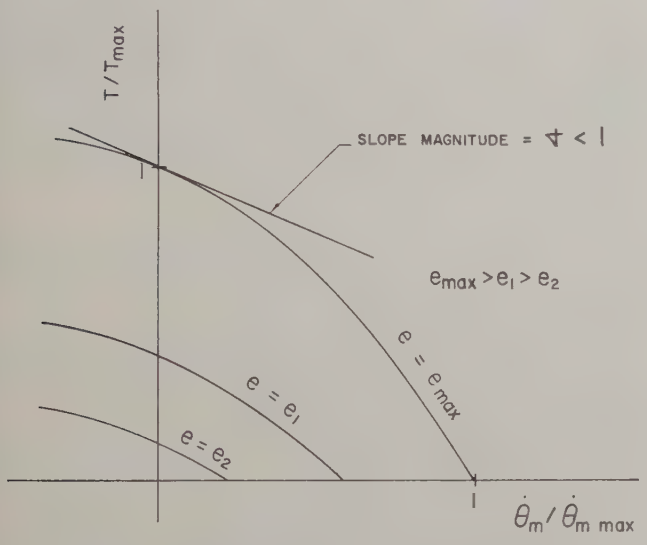


Fig. 2—Two-phase servomotor speed-torque curves.

The family defines a torque function $T = f(\dot{\theta}_m, e)$. Neglecting frictional torques

$$I\ddot{\theta}_m = f(\dot{\theta}_m, e), \tag{2}$$

where $I = I_m + I_g + I_t + I_p/n_g^2$. I_m is the motor inertia. This nonlinear differential equation may be linearized when $|\dot{\theta}_m| \ll \dot{\theta}_{m\ \max}$, which is frequently the case when the servo amplifier operates in the linear range; *i.e.*, when $|e| < e_s \approx e_{\max}$. Then (2) becomes

$$I\ddot{\theta}_m = -\gamma \frac{T_{\max}}{\dot{\theta}_{m\ \max}} \dot{\theta}_m + \frac{T_{\max}}{e_{\max}} e, \tag{3}$$

where γ, generally less than one in magnitude, is the slope parameter of the speed-torque curves defined in Fig. 2. Eq. (3) defines the motor transfer function as

$$Y_m(s) = \frac{K_m}{s(\tau_m s + 1)}, \tag{4}$$

where the constants K_m and τ_m are expressed in terms of the parameters of the speed-torque curve for $e = e_{\max}$ by

$$K_m = \frac{1}{\gamma} \frac{\dot{\theta}_{m\ \max}}{e_{\max}} = \frac{2\pi N_{\max}}{60\gamma e_{\max}} \frac{\text{rad}}{\text{volts sec}},$$

$$\tau_m = \frac{1}{\gamma} \frac{\dot{\theta}_{m\ \max} I}{T_{\max}} = \frac{2\pi N_{\max} I}{60\gamma T_{\max}} \text{ sec},$$

$$\frac{K_m}{\tau_m} = K_\infty = \frac{T_{\max}}{I e_{\max}} \frac{\text{rad}}{\text{volts sec}^2}. \tag{5}$$

$N_{\max}$ is the maximum motor velocity in rpm, and K_∞ is the motor gain constant for frequencies above the break frequency $1/\tau_m$. Expression of the transfer function parameters in the above form is important since it allows design equations to be expressed in terms of readily available speed-torque curve parameters.

The Position Servo

Block Diagram

Fig. 3 shows the block diagram for the position servo. The dimensionless input x and output y are selected so that the full-scale range on x and y is unity. The input compensation $Y_i(s)$ and the feedback compensation $Y_f(s)$ are defined to have unit magnitude at zero frequency; *i.e.*, $Y_i(0) = Y_f(0) = 1$. For series compensation, $Y_i(s) = Y_f(s)$. The nonlinear characteristics of the amplifier, motor, gear train, and potentiometer are not shown.

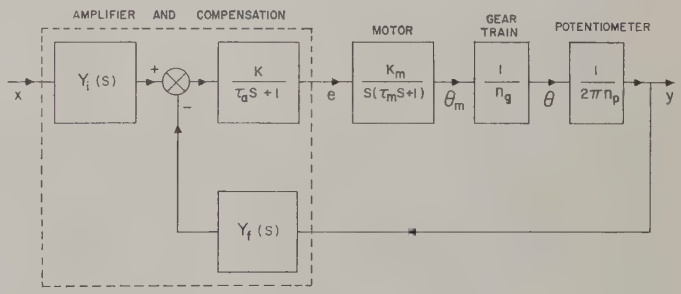


Fig. 3—Position servo block diagram.

Nonlinear Performance Requirements

A number of the performance requirements are nonlinear in nature and are particularly important in the selection of a motor and gear train. In this section, these requirements will be defined in terms of the component parameters. Later, their full significance in design will be considered.

Two requirements are the velocity and acceleration limits $\dot{y}_{\max}$ and $\ddot{y}_{\max}$. From the block diagram and definition of motor parameters

$$\dot{v}_{\max} = \frac{N_{\max}}{60 n_g n_p} \tag{6}$$

and

$$\ddot{v}_{\max} = \frac{T_{\max}}{2\pi n_g n_p I}. \tag{7}$$

In (6) and (7) the frictional torque is assumed small compared with T_{max}.

Another parameter that may be important is y_0, the first overshoot magnitude that results from a large step input. Exact determination of y_0 is difficult, but y_0 may be estimated as follows. Assume that servo amplifier inputs not causing saturation are small compared to y_0, that the torque reverses to full negative value when $\epsilon = x - y = 0$, that the deceleration to zero velocity is constant at $\ddot{y}_{max}$, and that velocity when $\epsilon = 0$ is $\dot{y}_{max}$. Then $2\,\ddot{y}_{max}\,y_0 = \dot{y}^2_{max}$. For an actual system, deceleration (proportional to torque in the negative velocity range in Fig. 2) may be somewhat greater than $\ddot{y}_{max}$, which occurs for $T = T_{max}$; also, for typical damping characteristics and linear range, torque reversal is earlier. Thus, a somewhat smaller y_0 than predicted results and

$$y_0 < \frac{\dot{y}^2_{max}}{2\ddot{y}_{max}} = \pi \frac{\left(\frac{N}{60}\right)^2 I}{n_g n_p T_{max}}. \tag{8}$$

Determination of the magnitude and number of successive overshoots for a large step input is difficult unless extreme simplifications are made. However, if y_0 is less than several times the amplifier input causing saturation, few excursions result with reasonable system damping. For a given motor, y_0 is decreased by increasing n_g or n_p. If I is considered to be independent of n_g (a reasonable approximation, typically) this means lower $\dot{y}_{max}$ and $\ddot{y}_{max}$.

Since frictional torque exists, there is a static error corresponding to motor torques less than the frictional torque referred to the motor shaft. Dynamically this error is observed as jumps in the output shaft position for a slowly varying input. To reduce this error to allowable values, a minimum gain is required. Thus, if δ is the allowed static resolution error,

$$\frac{T_{max}}{e_{max}} K\delta \geq T_T + \frac{T_p}{n_g}, \tag{9}$$

where $T_T = T_g + T_t$, the frictional torques acting at the motor shaft. Solving as an equality yields the minimum gain for a specified static resolution

$$K_{min} = \frac{1}{\delta} \frac{T_T + \frac{T_p}{n_g}}{T_{max}} e_{max}. \tag{10}$$

When tachometer damping is not used, T_T is small and the approximation

$$K_{min} = \frac{1}{\delta} \frac{1}{n_g} \frac{T_p}{T_{max}} e_{max}, \qquad T_T \ll \frac{T_p}{n_g} \tag{11}$$

is valid. It will be seen that K is limited by amplifier bandwidth, and therefore, increasing n_g is the only way of meeting a high resolution requirement for given potentiometers and motor.

Selection of the Gear Reduction Ratio

Evaluation of the above performance figures requires the gear reduction ratio. This ratio should be selected to optimize the performance in some sense.[1] Here some of the different factors will be considered and a minimum gear reduction ratio will be determined.

Frequently, the gear ratio is chosen to maximize the acceleration $\ddot{y}_{max}$ subject to the torque limitation T_{max}. If the frictional torques are neglected, this leads to the familiar result

$$n_g = \sqrt{\frac{I_p}{I_m + I_t + I_g}}. \tag{12}$$

Often this yields a small n_g. In fact, n_g may be such that the motor torque T_{max} is less than $T_T + T_p/n_g^2$, the frictional torque, so that the assumption of negligible frictional torques is certainly not valid.

Neglecting T_T but including T_p gives maximum $\ddot{y}_{max}$ when

$$n_g = \frac{T_p}{T_{max}} + \sqrt{\left(\frac{T_p}{T_{max}}\right)^2 + \frac{I_p}{I_m + I_t + I_g}}. \tag{13}$$

In many cases the inertia ratio in (13) is small compared with the squared torque ratio and $n_g \approx 2\,T_p/T_{max}$. Thus the maximum motor torque is approximately twice the frictional torque at the motor shaft.

To avoid interaction between the effects of frictional torque and saturation, the frictional torque should be small compared to available motor torque. The interaction, if allowed to exist, detracts from smooth tracking which is important in accurate analog computation, and the gear ratio should be selected accordingly. Experience has shown that maximum motor torque should exceed the frictional torque by a factor of at least five. Thus

$$T_{max} > 5\left(T_T + \frac{T_p}{n_g}\right) \tag{14}$$

or

$$n_g > \frac{T_p}{\frac{T_{max}}{5} - T_T}. \tag{15}$$

When tachometer damping is not used, T_T is small and (15) becomes

$$n_g > 5\,\frac{T_p}{T_{max}}, \qquad T_T \ll \frac{T_p}{n_g}. \tag{16}$$

[1] The following authors have determined motor suitability and the gear reduction ratio, considering $\dot{y}_{max}$ and $\ddot{y}_{max}$ requirements and frictional torque limitations but neglecting the other performance figures and their relation to linear system design. H. Harris, "A comparison of two basic servomechanism types," *Trans. AIEE*, vol. 66, pt. II, pp. 83–93; 1947. G. C. Newton, Jr., "What size motor?" *Machine Design*, vol. 22, pp. 125–130; November, 1950.

Inequality (15) or (16) frequently sets a minimum gear ratio that is considerably greater than the gear ratio for maximum acceleration, in which case the potentiometer inertia I_p is not significant in design.

If $\dot{y}_{max}$, $\ddot{y}_{max}$, and good tracking characteristics were the only performance figures, (12), (13), (15), or (16) would set the minimum gear ratio for the constraint of a given motor and potentiometer. However, requirements on y_0 or static resolution may require an increased minimum gear ratio. Thus (8) bounding y_0 may require higher n_g to obtain an acceptable value of y_0. Similarly, specific K_{min} and δ may require higher n_g to satisfy (11). Linear analysis will show that K_{min} is indeed limited by the servo amplifier bandwidth.

With the above limitations in mind, a minimum gear ratio may be determined for specified motor and potentiometers. If the requirements for velocity and acceleration limits are not met with this n_g, a motor with greater maximum torque and/or velocity must be selected, and a new set of performance figures must be computed.

Compensation of the Linear System

To assure smooth tracking, the gear ratio has been selected so that motor torque exceeds frictional torque by a factor of at least five. This means that there is at least a range of five-to-one where motor torque can exceed frictional torque and be proportional to amplifier input. In this range, system design on a linear basis will yield reasonable results. In this section different compensation techniques to achieve specified linear characteristics such as bandwidth, damping, and steady-state errors will be considered. In certain instances, fundamental limitations on the characteristics exist and may influence the selection of components and the gear reduction ratio. These limitations will be apparent from expressions relating the component parameters and linear and nonlinear performance figures.

First, consider series compensation where $Y_i = Y_f$ is the series compensating function. The closed-loop transfer function $Y(s)$ relating the Laplace transform of y to that of x is then

$$Y(s) = \frac{Y_i}{Y_f}\frac{Y_0}{1+Y_0} = \frac{Y_0}{1+Y_0}, \tag{17}$$

where

$$Y_0(s) = Y_f \frac{KK_m}{2\pi n_g n_p s(\tau_m s+1)(\tau_a s+1)}. \tag{18}$$

For simplicity, perfect error rate or lead compensation,

$$Y_f = Y_i = 1 + C_e s, \tag{19}$$

will be used. Imperfect and physically realizable compensation yields similar results.

Three cases will be analyzed. The first and most common occurs when $25/\tau_m < 1/\tau_a$. The second and third cases are defined by $5/\tau_m < 1/\tau_a < 25/\tau_m$ and $1/\tau_m < 1/\tau_a < 5/\tau_m$. Analysis will be based on plots of decibel magnitude of Y_0/K vs log ω, with compensation selected so that Y_0 has a slope of approximately -9 db/octave when $|Y_0| = 1$. This will give a phase margin of about 45° and, hence, a reasonable resonant peak in the closed-loop transfer function magnitude. The peaking frequency ω_p will be estimated as the frequency where $|Y_0| = 1$. The accuracy of this procedure will be sufficient to determine the required interrelationships between system parameters.

Fig. 4 shows the first case where C_e has been selected to maximize K. Here

$$\frac{1}{C_e} = \frac{1}{5}\frac{1}{\tau_a}, \qquad \omega_p = \frac{3}{5}\frac{1}{\tau_a}, \qquad \frac{1}{C_e} > 5\frac{1}{\tau_m} \text{ since } \frac{1}{\tau_a} > 25\frac{1}{\tau_m}. \tag{20}$$

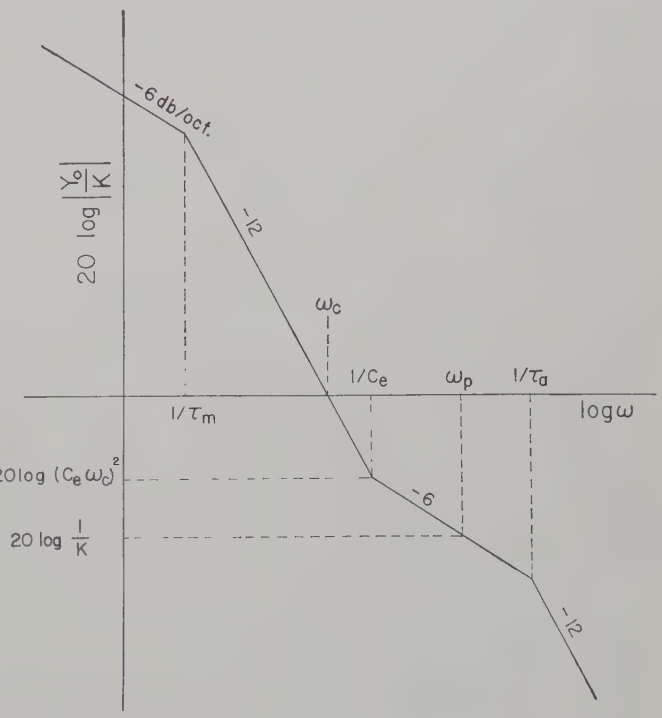


Fig. 4—Open-loop gain for position system with error rate compensation when $25/\tau_m < 1/\tau_a$.

Since $\omega_p = 3/C_e$, it is apparent from Fig. 4 that

$$\frac{1}{K} = \frac{1}{3}(C_e\omega_c)^2, \tag{21}$$

where

$$\omega_c = \sqrt{\frac{K_m}{2\pi n_g n_p \tau_m}} = \sqrt{\frac{K_\infty}{2\pi n_g n_p}}. \tag{22}$$

Substituting (22) and (20) into (21) gives

$$K = \frac{3}{25}\frac{2\pi n_g n_p I e_{max}}{\tau_a^2 T_{max}}. \tag{23}$$

For given motor, potentiometer, amplifier, and gear train, this is the maximum amplifier gain.

If the above K value equals or exceeds $K_{\min}$ determined by (10) or (11), then the specified resolution δ is obtained. If K does not exceed $K_{\min}$, the resolution requirement cannot be met unless certain system parameters are changed. The interrelationship of these parameters is most easily seen by substituting in $K > K_{\min}$ from (23) and (10) leading to the inequality

$$\frac{1}{\tau_a} > \sqrt{\frac{25}{6\pi} \frac{T_T + \frac{T_p}{n_g^2}}{n_g n_p I \delta}} \tag{24}$$

or

$$\frac{1}{\tau_a} > \frac{1}{n_g} \sqrt{\frac{25}{6\pi} \frac{T_p}{n_p I \delta}}, \qquad T_T \ll \frac{T_p}{n_g}. \tag{25}$$

These inequalities are useful in selecting components that are compatible with specified static resolution. For example, if potentiometers, motor, and amplifier are given, (24) determines a minimum n_g for specified δ. This n_g may well very exceed that given in (13) or (15). The advantage of high amplifier bandwidth and low-frictional torques is clear. It is also clear that nonlinear characteristics cannot be divorced from the linear system design.

The above discussion concerns the limitations placed on gain and static resolution by the compensation restrictions imposed by amplifier bandwidth; the converse is sometimes true. For example, suppose 2δ equals the resolution range $1/\eta$ of the reference potentiometer; certainly 2δ cannot be less than the wire-to-wire jump $1/\eta$. In this case $K_{\min}$ also becomes a maximum $K = K_{\max}$ since larger K could result in wire-to-wire hunting, causing excessive potentiometer wear. That is to say, an error corresponding to one-half of a potentiometer wire can produce torque exceeding frictional torque when K is greater than $K_{\min} = K_{\max}$. When $K_{\max}$ is less than that given by (23), the compensation of Fig. 4 must be changed. By increasing C_e over the value of (20), K can be decreased while ω_p is maintained near $1/\tau_a$.

In the above work, it was assumed that $25/\tau_m < 1/\tau_a$. For relatively low speed, high torque to inertia motors with $\gamma \approx 1$, τ_m given by (5) is small, and the inequality may not be valid, thus requiring a modification of the results. For $5/\tau_m < 1/\tau_a < 25/\tau_m$, Fig. 4 is still correct in form, but ω_p and K may be increased somewhat because of the additional phase lead contributed by the motor transfer function. Suppose, for example, that $5/\tau_m \approx 1/\tau_a$; then K may be increased so that $\omega_p \approx 1/\tau_a$ and the three is replaced by five in (21) and (23). In (24) and (25), the factor 25/6 is replaced by 5/2.

When $1/\tau_m < 1/\tau_a < 5/\tau_m$, a different compensation must be used to maximize K and ω_p. This is shown in Fig. 5 where

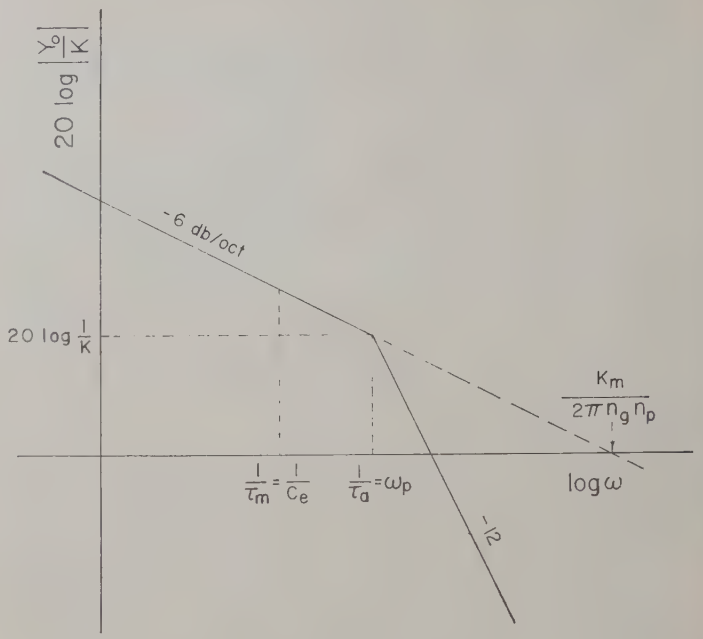


Fig. 5—Open-loop gain for position system with error rate compensation when $1/\tau_m < 1/\tau_a < 5/\tau_m$.

$$\frac{1}{C_e} = \frac{1}{\tau_m},$$
$$\omega_p = \frac{1}{\tau_a}, \tag{26}$$

and

$$K = \frac{\frac{1}{\tau_a}}{\frac{K_m}{2\pi n_g n_p}} = n_g n_p \frac{\gamma}{\tau_a} \frac{60}{N_{\max}} e_{\max}. \tag{27}$$

Again using $K > K_{\min}$, it is found that

$$\frac{1}{\tau_a} > \frac{1}{n_g n_p} \frac{N_{\max}}{60\gamma} \frac{T_T + \frac{T_p}{n_g}}{T_{\max}} \frac{1}{\delta}, \tag{28}$$

or

$$\frac{1}{\tau_a} > \frac{1}{n_g^2 n_p} \frac{N_{\max}}{60\gamma} \frac{T_p}{T_{\max}} \frac{1}{\delta}, \qquad T_T \ll \frac{T_p}{n_g}. \tag{29}$$

An important figure of merit in linear system design is the steady-state error for ramp inputs. In determining this error, it is useful to take the following more general, physically realizable forms for Y_i and Y_f:

$$Y_i = 1 + \frac{C_e s}{\alpha_1 C_e s + 1},$$
$$Y_f = 1 + \frac{C_d s}{\alpha_2 C_d s + 1}. \tag{30}$$

For series compensation, $C_d = C_e$ and $\alpha_2 = \alpha_1$. Using the final value theorem gives the linear-system steady-state error

$$\epsilon_l = \left[\frac{2\pi n_0 n_p}{KK_m} + C_d(1-\alpha_2) - C_e(1-\alpha_1)\right]V \quad (31)$$

for the input $x = Vt$. For series compensation the last two terms cancel. In certain input circuitry, Y_i and Y_f can be adjusted separately, and C_e and α_1 may be selected to make $\epsilon_l = 0$. This does not change the previous results on series compensation since Y_i and Y_f will differ little for $\epsilon_l = 0$.

Zero steady-state error for the ramp input is equivalent to zero slope of phase vs frequency at zero frequency in the closed-loop system. This characteristic is particularly desirable in servos used for analog computation. However, in many cases the first term of (31) is small and cancellation is not warranted.

In addition to the linear-system error, there is the nonlinear steady-state error,

$$\epsilon_n = \frac{V}{|V|}\delta, \quad (32)$$

for the input $x = Vt$. Thus an error equal to δ must exist before motor torque is available to exceed frictional torque. Since this error is independent of $|V|$, it cannot be cancelled out for all V.

Derivative control through tachometer feedback is another commonly used form of compensation. In this case

$$Y_i = 1,$$
$$Y_f = 1 + \frac{\tau s}{\tau s + 1} C_d s, \quad (33)$$

where the high-pass filter $\tau s/(\tau s+1)$ in the tachometer path is used to eliminate the steady-state error for ramp inputs [the second term in (31)]. For $\tau > C_d$ ($\tau \approx C_d$ usually gives acceptable damping) the approximation

$$Y(j\omega) \approx 1 + C_d j\omega, \qquad \omega > \frac{1}{C_d} \quad (34)$$

is useful since $Y(j\omega)$ then becomes

$$Y(j\omega) = \frac{Y_i}{Y_f}\frac{Y_0}{1+Y_0} \approx \frac{1}{1+C_d j\omega}\frac{Y_0}{1+Y_0}, \qquad \omega > \frac{1}{C_d}. \quad (35)$$

The factor $1/(1+C_d j\omega)$ gives increasing attenuation for $\omega > 1/C_d$ so that the $|Y(j\omega)|$ may not exhibit great peaking even though $|Y_0|/|1+Y_0|$ does become large. This is apparent from Fig. 6 where the plots of decibel gain vs log ω for Y_0/K and Y are shown for a value of K much greater than that given by (23). The phase margin for Y_0 is now very small, resulting in a large peaking in $|Y_0|/|1+Y_0|$ at $\omega = \omega_p$. However, $|Y|$ does not greatly exceed one at $\omega = \omega_p$, because of the factor $1/(1+C_d j\omega)$. Evidently, the derivative control allows greater gain than series compensation. The useful bandwidth is $1/C_d$.

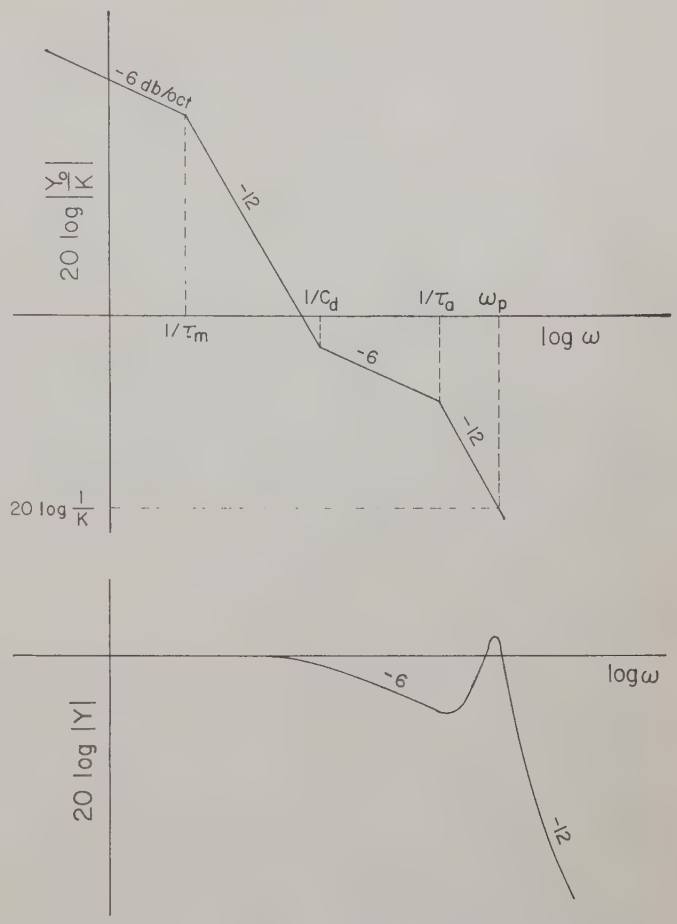


Fig. 6—Open and closed-loop gain for position system with tachometer compensation.

The gain advantage of derivative control would indicate that static resolution was not limited by amplifier bandwidth for this form of compensation. Unfortunately, large increase of gain is not usually possible because of additional time lags neglected in the analysis. Such lags could reduce the phase margin to a negative value resulting in an unstable system. The maximum increase in K through derivative control can be obtained by a more complete analysis considering the additional time lags and using an inverse Nyquist plot. It is difficult to generalize on the possible gain increase due to the wide variation of neglecting time lags, but a factor of two to ten above the gain for series compensation is not uncommon.

The increase in K does not necessarily mean a corresponding decrease in the static resolution error. This is apparent from (10) since the added frictional torque from the tachometer requires increased K to maintain a given δ.

In addition, the added tachometer inertia reduces the acceleration limit. The lower cost and great reliability of series RC compensation may also outweigh any advantage obtained with tachometer damping.

Another means of increasing K above the value given by (23) is by using a series lead-lag compensating function

$$Y_i = Y_f = \left(\alpha_1 \frac{1 + \dfrac{C_i}{s}}{1 + \dfrac{\alpha_1 C_i}{s}} \right) \left(\frac{1 + C_d s}{1 + \alpha_2 C_d s} \right). \qquad (36)$$

The second factor approximates the error rate control; the first factor approximates integral control and permits an increase in K by a factor of about $1/\alpha_1$ for correctly chosen C_i. Such compensation has two main disadvantages. First, the compensating network is more complicated and may require excessively large capacitor values for realization. Second, lag action introduces additional open-loop negative phase shift that may cause a nonlinear instability or long recovery time for inputs causing amplifier saturation. For these reasons, such compensation has limited use.

The Velocity Servo

Block Diagrams

Fig. 7 shows the block diagram for the velocity servo. A tachometer voltage proportional to the motor shaft velocity is the feedback so that shaft velocity is ideally proportional to input voltage. The tachometer is mounted on the motor shaft to eliminate gear backlash in the feedback. The dimensionless output y is again selected so that the full-scale range on y is unity. The input and feedback compensation are defined to have unit gain at zero frequency; *i.e.*, $Y_i(0) = Y_f(0) = 1$. K_i is chosen so that the output velocity $\dot{y}$ is proportional to the input voltage e_i by the desired factor.

For design purposes, it will be useful to consider the equivalent block diagram of Fig. 8 where u is a dimensionless input voltage and

$$K_v = \frac{K_i}{60 n_g n_p K_t}. \qquad (37)$$

System Design

Much of the previous work applies for the velocity system. Thus, (6) and (7) still give correct values for $\dot{y}_{max}$ and $\ddot{y}_{max}$. y_0 no longer has significance. The same approach for gear reduction ratio determination is valid and (12), (13), or (15) gives the minimum value for n_g.

The velocity system has a velocity resolution error instead of a position error. That is, the velocity error $\epsilon = u - \dot{y}$ must exceed a certain value before the motor torque exceeds frictional torque. This velocity resolution error δ_v is given by

$$\delta_v = \frac{T_T + \dfrac{T_p}{n_g}}{\dfrac{T_{max}}{e_{max}} K K_t \dfrac{60}{2\pi}}. \qquad (38)$$

K must be large enough so that δ_v is sufficiently small. It will be seen that there is no particular limit on K as there was previously.

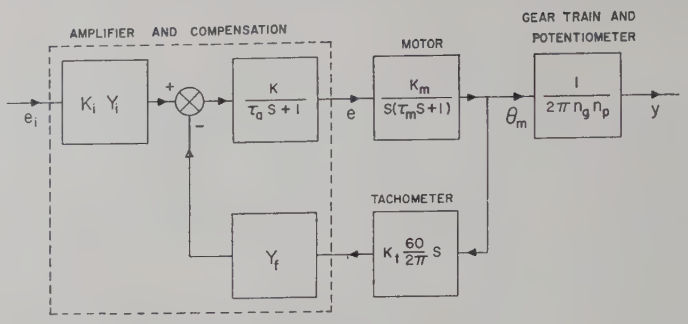


Fig. 7—Velocity servo block diagram.

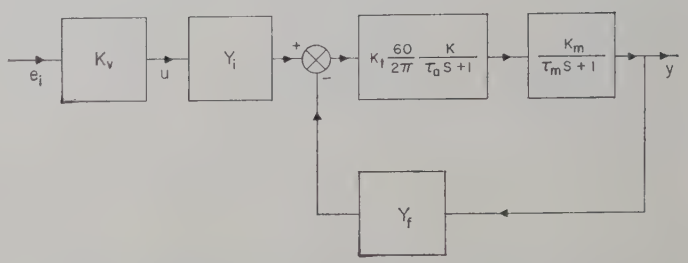


Fig. 8—Equivalent block diagram for velocity servo.

Since the velocity servo is a type-zero system, it has a steady-state linear-system error for $u = U$, a constant. The error is given by

$$\epsilon_l = \frac{U}{1 + \dfrac{60}{2\pi} K_t K K_m} = \frac{U}{\dfrac{60}{2\pi} K_t K K_m} = \frac{\gamma e_{max}}{K_t K N_{max}} U, \quad 1 \ll \frac{60}{2\pi} K_t K K_m. \qquad (39)$$

In addition, there is a nonlinear resolution error

$$\epsilon_n = \frac{U}{|U|} \delta_v. \qquad (40)$$

Typically, $|\epsilon_n| \gg |\epsilon_l|$. In fact, when $U = \dot{y}_{max}$, ϵ_n is usually larger than ϵ_l. Thus design for specified δ_v usually assures steady-state errors not greatly exceeding δ_v.

The compensation of the velocity servo is easier than the position system since the open-loop system no longer has a $1/s$ term present. The main problem is to achieve sufficient gain to meet a requirement on δ_v. Frequently this can be done with a simple series lead compensation network. The more complicated compensation given by (36) may be necessary. In this case there is no stability problem for saturating inputs because of the simpler open-loop characteristics.

An Example

To illustrate the above principles, a particular design will now be considered. The problem is to design a position servo with five multiplying potentiometers. Line frequency and the choice of a ten-watt magnetic ampli-

fier designate a 60-cps, two-pole, 4-ounce inch servomotor. The potentiometer linearity requirement is 0.1 per cent and the static resolution error is 0.05 per cent of full scale. For simplicity and reliability, a simple series RC lead compensating network is to be used. Under these restrictions, the servo is to be designed for the maximum possible velocity and acceleration limits and bandwidth.

Examination of (6) and (7) shows that for a given motor, highest $\dot{y}_{max}$ and $\ddot{y}_{max}$ are obtained for smallest n_p. This dictates single revolution potentiometers and an $n_p=1$. Consistent with reliability, cost, and similar considerations, a six-gang potentiometer is selected for minimum T_p and $\eta>1000$. Ganged 20,000-ohm, wirewound potentiometers with the following characteristics meet the specifications.

$T_p=4$ ounce inches $=2.08\cdot10^{-2}$ pound feet,
$I_p=120$ grams cm$^2=8.8\cdot10^{-4}$ slug feet^2,
$\eta=1800$.

I_p includes the inertia of the potentiometer coupling.

The high torque-to-inertia-ratio servomotor has

$T_{max}=4$ ounce inches $=2.08\cdot10^{-2}$ pound feet,
$N_{max}=3400$ rpm,
$e_{max}=115$ volts,
$\gamma=0.8$,
$I_m=5$ grams cm$^2=3.66\cdot10^{-7}$ slug feet^2.

The maximum power input per phase is 9 watts, so the magnetic amplifier has sufficient power capability.

The gear train has $T_g\ll4$ ounce inches and $\zeta\ll1/\eta$. For a fairly large range in n_g, the gear train inertia referred to the motor shaft is

$$I_g = 2 \text{ grams cm}^2 = 1.47\cdot10^{-7} \text{ slug feet}^2.$$

Design for maximum acceleration, neglecting frictional torque, is given by (12) yielding

$$n_g = 4.1.$$

Consideration of frictional torque using (13) gives

$$n_g = 5.3.$$

The minimum T_{max} to T_p/n_g ratio of five requires

$$n_g > 5.$$

Thus, design for maximum acceleration considering frictional torque is possible, neglecting resolution limitations.

To determine the effect of the resolution requirement, it is first necessary to compare $1/\tau_a$ and $1/\tau_m$. With current feedback, the magnetic amplifier has $1/\tau_a=125$ rad/sec. By (5)

$$\frac{1}{\tau_m} = 91\,\frac{1}{1+\dfrac{17.1}{n_g^2}},$$

and for $n_g\geq5.3$, $1/\tau_m<1/\tau_a<5/\tau_m$. Thus (29) is appropriate; substitution of the specified $\delta=5\cdot10^{-4}$ and other parameters gives the inequality

$$n_g > 33.6.$$

Hence, the linear system design and resolution specification require a much larger n_g than that for maximum acceleration. For $n_g<33.6$, it is not possible to achieve design resolution and acceptable stability. Setting $n_g=35$ gives

$$C_e = \tau_m = \frac{1}{89.5} \approx 0.01;$$

and, by (27),

$$K = 7100.$$

Computing of performance figures yields

$$\dot{y}_{max} = 1.62,$$
$$\ddot{y}_{max} = 182,$$
$$y_0 < 0.0072.$$

For a ramp input $x=Vt$, the steady-state linear-system and nonlinear errors are given by (31) and (32).

$$\epsilon_l = 0.008V$$
$$\epsilon_n = 0.0005\,\frac{V}{|V|}.$$

The linear-system error may be cancelled by designing the input circuitry so Y_i and Y_f differ by an appropriate amount.

Correspondence

Orthogonal Matrices, Error-Correcting Codes and Load-Sharing Matrix Switches*

The purpose of this note is 1) to point out the application of the theory of orthogonal matrices[1] to the construction of multiple-error-correcting codes[2,3] and to the design of very efficient load-sharing matrix switches[4,5] and 2) to extend the theory further to the case of pseudo-orthogonal matrices and their applications.

The theory of orthogonal matrices was investigated by Paley in connection with the theory of polytopes. He studied square matrices of order n by n where the elements are two-valued variables. Therefore, it is possible to assume these as variables which take the value of either $+1$ or -1. The orthogonality condition requires that if any two rows are compared, the number of similarities equals the number of dissimilarities. In the language of error-correcting codes, each row may be viewed as a sequence and the condition is that the distance between any pair of sequences is $n/2$. Therefore, an orthogonal matrix, if properly interpreted, actually gives an error-correcting code of size n with distance $n/2$.

In order to construct an error-correcting code of $2n$ sequences with minimum distance $n/2$, it is only necessary to complement the matrix. The reason that the matrix with its complement form a code as specified is clear.

Example 1: Given an orthogonal matrix of order 4

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

it is possible to obtain a code of 8 members with a minimum distance of 2. The code is the following.

1	1	1	1	1	1	1	1
1	−1	1	−1	1	0	1	0
1	1	−1	−1	1	1	0	0
1	−1	−1	1	1	0	0	1
−1	−1	−1	−1	0	0	0	0
−1	1	−1	1	0	1	0	1
−1	−1	1	1	0	0	1	1
−1	1	1	−1	0	1	1	0

In the application of these matrices to the design of load-sharing matrix switches, only a slight modification is necessary. Since in each core of the matrix switch the number of wires wound in one direction must equal the number of wires wound in the other direction, the number of outputs is $n-1$ for a matrix of order n.

The design methods by Constantine[4] and Marcus[5] usually result in switches of low efficiency as measured in terms of the numbers of input required for a given number of outputs. The use of the theory of orthogonal matrices improves the situation in general. The efficiency of switches using three methods of design is compared for a set of required outputs as shown in Table I. It is clear that great savings are possible.

TABLE I

Number of Outputs Required	Number of Inputs Required				Noise / Signal*
	Constantine	Marcus	Method using Orthogonal Matrix	Method Using Orthogonal and Pseudo-Orthogonal Matrices	
5	16	8	8	6	33.3%
8	16	16	12	12	0%
16	32	32	20	18	11.1%
32	64	64	36	36	0%
36	128	64	40	38	5.2%
64	128	128	68	68	0%
72	256	128	76	74	2.8%

* This column gives the ratio of maximum non-desirable output to desirable output if the fifth column is used

The weak point of Marcus's method is that it has a low efficiency when n is a power of 2. Therefore, it fails to improve the efficiency over Constantine's method in the very important case of binary computers. The new approach has no such difficulty.

One of the necessary conditions for a matrix to be orthogonal is for n to be a multiple of 4. However, if pseudo-orthogonal matrices are allowed, solution can be obtained for a large set of $n=2$ (mod. 4). A pseudo-orthogonal matrix is characterized by the property

$$\frac{n}{2} - 1 \leq d \leq \frac{n}{2} + 1,$$

where d is the distance between any pair of rows. This result is an outgrowth of Plotkin's work.

Consider a function $a(k)$ where the independent variable k takes integer value $0, 1, \cdots, p-1$. The following definition is made for $a(k)$:

$$a(0) = 1$$
$$a(k) = \left(\frac{k}{p}\right)$$

where (k/p) is the Legendre symbol. (k/p) takes the value $+1$ or -1, according to whether k is a quadratic residue or a quadratic nonresidue of prime p. Consider the sequence

$$S_0 \quad a(0), a(1), \cdots a(p-1),$$

and its $(p-1)$ cyclic permutations. The distance between any pair of sequences is known if one knows the distance between any sequence S_k and S_0.

$$S_0 \quad a(0)\ a(1) \quad a(2) \quad a(3) \cdots a(p-1)$$
$$S_k \quad a(k)\ a(k+1)\ a(k+2)\ a(k+3) \cdots a(p+k-1).$$

Utilizing the primitive root ϵ of p it is sufficient, after an obvious rearrangement, to consider the following sequences

$$a(0)\ a(\epsilon^r) \quad a(2\epsilon^r) \cdots a(-\epsilon^r)$$
$$a(\epsilon^r)\ a(2\epsilon^r)\ a(3\epsilon^r) \cdots a(0)$$

where r is an integer.

The property

$$\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) = \left(\frac{ab}{p}\right)$$

holds for the Legendre symbols.[6] Thus, the distance is obtainable from comparing the following two sequences:

$$a(0) \quad a(1) \quad a(2) \quad a(3) \cdots a(-\epsilon^r)$$
$$a(\epsilon^r) \quad a(2) \quad a(3) \quad a(4) \cdots a(0).$$

When k is a quadratic residue, r is even, and the end points do not contribute to the distance of the two sequences. The distance obtained here shall be called D. When k is a quadratic nonresidue, r is odd and each of the end points contributes a distance one to the total distance. Thus, the total distance is $D+2$. From this fact and a counting procedure which calculates the distance contributed by each column of the entire set of p sequences, it is possible to deduce that

$$D = \frac{p-1}{2}.$$

If a -1 is added at the end of each permutation and the sequence of all ones is added to the p augmented permutations, a pseudo-orthogonal matrix is obtained. Furthermore, the procedure just described is actually a method of construction of error-correcting code of size $2n$ with distance

$$\frac{n}{2} - 1 \leq d \leq \frac{n}{2} + 1$$

if all 1's are changed into zeros and the whole thing is complemented. It is clear that the construction of the pseudo-orthogonal matrix and the application to load-sharing matrix switch design is possible for any $n=2$ (mod. 4), if $n-1$ is a prime. Some further saving is possible as is shown in the last column of Table I. However, some output will also exist in cores other than the selected one. The ratio of undesirable output to desired output is inversely proportional to n. Therefore, it can be tolerated in most practical cases.

ROBERT T. CHIEN
IBM Research
Yorktown Heights, N. Y.

* Received by the PGEC, May 14, 1959.

[1] R. E. A. Paley, "On orthogonal matrices," *J. Math. Phys.*, vol. 12, pp. 311–320; 1933.

[2] R. T. Chien, "On the characteristics of error-correcting codes," IRE TRANS. ON INFORMATION THEORY, vol. IT-5, p. 91; June, 1959.

[3] M. Plotkin, "Binary Codes with Specified Minimum Distance," Res. Div., Univ. of Pennsylvania, Philadelphia, Rept. 51–20; January, 1951.

[4] G. Constantine, "A load sharing matrix switch," *IBM J. Res. & Dev.*, vol. 2, pp. 204–211; July, 1958.

[5] M. Marcus, "Doubling the efficiency of the load-sharing matrix switch," *IBM J. Res. & Dev.*, vol. 3, pp. 194–196; April, 1959.

[6] R. Carmichael, "The Theory of Numbers," John Wiley and Sons, Inc., New York, N. Y.; 1914.

Treatment of Transition Signals in Electronic Switching Circuits by Algebraic Methods*

Use of Boolean algebra in the design of electronic switching circuits is based upon the assumption that signals in the circuit have reached their end points, or limiting values. Signals at these end points are designated as either 0 or 1, depending upon the end point, and the rules of Boolean algebra permit the investigation of circuits which behave in equivalent ways for such signals. It is interesting to note, however, that many of these rules may continue to be applicable even while signals are undergoing transition between the end points 0 and 1. By using identities based upon this slightly more limited set of rules, we can investigate circuits which remain "equivalent" even during transition times.

The algebraic system to be treated involves variables $a, b, c, \cdots, x$, which may be normalized to lie in the range $0 \leq x \leq 1$. Switching elements of the AND and OR type generally produce minimum and maximum functions, respectively, when intermediate signals are applied. Thus we shall define:

$$\begin{aligned} &\text{AND:} \quad a\,b = \min\,[a, b] \\ &\text{OR:} \quad a \vee b = \max\,[a, b]. \end{aligned} \tag{1a}$$

A system involving just these two operations is a distributive lattice with a 0 and a 1, and any Boolean identity not involving complementation is valid in such a system [1]. Some applicable identities are listed in (2a).

For any a, b, and c in the system

$$\begin{aligned} a\,a &= a \\ a\,b &= b\,a \\ a\,(b\,c) &= (a\,b)c \\ a\,(b \vee c) &= a\,b \vee a\,c \\ a\,(b \vee a) &= a \\ 1\,a &= a \\ 0\,a &= 0 \\ a \vee a &= a \\ a \vee b &= b \vee a \\ a \vee (b \vee c) &= (a \vee b) \vee c \\ a \vee b\,c &= (a \vee b)(a \vee c) \\ a \vee b\,a &= a \\ 0 \vee a &= a \\ 1 \vee a &= 1. \end{aligned} \tag{2a}$$

Many other valid identities may be constructed. In questionable cases one may always check the validity of an identity by the method of total induction, *i.e.*, the substitution of all combinations of 0's and 1's for the variables. This check may still be used in spite of the fact that intermediate signals are permitted, as long as the operation of complementation appears nowhere in the identity.

The inclusion relation $a \subseteq b$, may be defined by either $ab = a$ or $a \vee b = b$. An interpretation of $a \subseteq b$ in our present system is that signal a is less than or equal to signal b at all times under consideration.

Use of the switching element NOT produces the logical operation of complementation. The interpretation of this operation for intermediate signals is somewhat more difficult. It is generally possible to assume, however, that it represents a monotone decreasing function of the input. Thus we shall define:

$$\text{NOT:}\ \bar{a} = f(a), \text{ where } f(a) \text{ is some function having the property that } f(0) = 1, f(1) = 0, \tag{1b}$$

and whenever $a \leq b$ we have $f(a) \geq f(b)$.

By combining (1a) with (1b) we may include DeMorgan's laws in our list of identities.

$$\overline{(a\,b)} = \bar{a} \vee \bar{b} \qquad \overline{(a \vee b)} = \bar{a}\,\bar{b}. \tag{2b}$$

Not all formulas of Boolean algebra remain valid, however, and, in particular, the rules $a\bar{a} = 0$ and $a \vee \bar{a} = 1$ must be replaced by the weaker form given in (2c).

$$a\,\bar{a} \subseteq b \vee \bar{b}. \tag{2c}$$

In view of our definition of the inclusion relation we may rewrite (2c) as either of the identities $a\bar{a}(b \vee \bar{b}) = a\bar{a}$ and $a\bar{a} \vee b \vee \bar{b} = b \vee \bar{b}$.

Some advantage is also achieved by replacing the general assertion of (1b) by the specific definition.

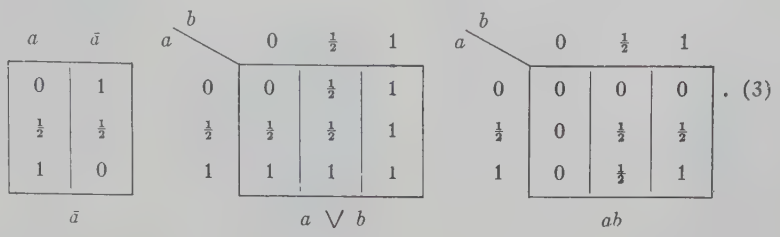
$$\text{NOT:} \quad \bar{a} = 1 - a. \tag{1c}$$

Not only do the previous rules (2a)–(2c) continue to hold, but we may now include the rule of involution

$$\bar{\bar{a}} = a. \tag{2d}$$

The applicability of (1c) and therefore (2d) to electronic switching elements is somewhat less reasonable than that of (1a) and (1b) since NOT elements generally contain amplification. Yet, in practical problems involving equivalences of circuits, rule (d) and its consequences are seldom involved, and in those cases in which they are, one is rarely likely to draw incorrect qualitative conclusions as a result.

The complete set (2a)–(2d) of relations represents a system only slightly weaker than Boolean algebra and formally equivalent to the strong 3-valued logic of Kleene [2]. In this logic, signal values may be written as 0, $\frac{1}{2}$, 1, giving the following operation tables:



. (3)

Any identity which is valid when all combinations of $\{0, \frac{1}{2}, 1\}$ are substituted for the variables will also be valid in general. Thus, (3) provides a test for all equivalences under (2a)–(2d).

Switching circuits which are equivalent in the sense of (3) should produce similar outputs even if the signals involved have not reached their end points. They are, therefore, more closely related than circuits which are merely equivalent in the Boolean sense. Circuits which are equivalent in the sense of (3) exhibit the same transient behavior and if one of two such equivalent circuits is "spike-free," the other will be also, and conversely.

Example 1: Show that the identity

$$a\,\bar{b} \vee \bar{a}\,b = (a \vee b)(\bar{a} \vee \bar{b})$$

holds in the system defined by (2a)–(2d).

Proof: In the nine cases $a = \{0, \frac{1}{2}, 1\}$ and $b = \{0, \frac{1}{2}, 1\}$ we have equivalence, so we conclude that the identity is valid in the entire square $0 \leq a \leq 1$ and $0 \leq b \leq 1$.

Example 2: Show that the Boolean identity

$$a\,b \vee \bar{a}\,c = a\,b \vee \bar{a}\,c \vee b\,c$$

does not hold in the system (2a)–(2 d).

Proof: We let $b = c = 1$ and let $a = \frac{1}{2}$. Then $\bar{a} = \frac{1}{2}$ and the left-hand side has value $\frac{1}{2}$, while the right-hand side has value 1.

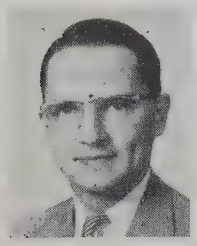
D. E. Muller
Digital Computer Lab.
University of Illinois
Urbana, Ill.

* Received by the PGEC, May 12, 1959.

References

[1] G. Birkhoff, "Lattice Theory," American Mathematical Society Publications, New York, N. Y., p. 133; 1948.
[2] S. C. Kleene, "Introduction to Metamathematics," D. Van Nostrand Co., Inc., Princeton, N. J., pp. 334; 1952.
[3] C. C. Elgot, "Lectures on Switching and Automata Theory," Univ. of Michigan, Ann Arbor; 1958.
[4] G. A. Metze, "Many-Valued Logic and the Design Switching Circuits," Master's thesis, Univ. of Illinois, Urbana; 1955.
[5] C. Y. Lee and W. H. Chen, "Several-valued combinational switching circuits," *Commun. & Electronics*, no. 25, (*Trans. AIEE*, pt. I, vol. 75), pp. 278–283; July, 1956.

Contributors



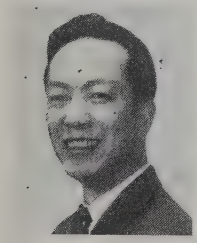
D. J. Blattner

Donald J. Blattner (SM'58) was born in New York, N. Y., on December 29, 1925. He received the B.S. degree in electrical engineering in 1946 and the M.S. degree in physics in 1949, both from Columbia University.

From 1949 to 1953, he was a teaching assistant and lecturer in physics at Columbia while doing research in the solid-state laboratory. Since 1953 he has been at RCA, Princeton, N. J., working on microwave tubes and circuits, and on telecommunication systems.

Mr. Blattner is a member of Sigma Xi.

❖



S. K. Chao

Stanley K. Chao (M'55) was born in Kunming, China, in November, 1924. He received both the B.S. and the M.S. degrees in electrical engineering in 1949 from Massachusetts Institute of Technology, Cambridge, Mass., and is presently studying toward his Ph.D. degree in applied mathematics at Harvard University in Cambridge.

During his last two years of school he worked as a test engineer for the General Electric Company under the cooperative plan of M.I.T., and from 1949 to 1950 he was a control designer for the High-Voltage Engineering Corporation in Burlington, Mass. From 1950 to 1953, he worked with Baird Associates, Inc., in Cambridge, Mass., where he participated in the engineering and design of spectroscopic instruments. He began work with Baird-Atomic, Inc., also in Cambridge, in 1953, where he was made head of the data processing group responsible for development and design of data processing equipment. During 1955 and 1956 he was the associate project engineer on the Model 580 digital correlator, a special-purpose computer that has been in operation at the Naval Research Laboratory since December, 1956. Other projects he has undertaken include a data reduction system for wave propagation study, a multichannel accumulator readout for the Naval Research Laboratory, a data processing system for the National Bureau of Standards, and a machine control system for General Electric Company. He joined Sylvania Electric Products, Inc. in Needham, Mass., in 1958 and is presently a section head in the Data Processing Laboratory, responsible for advanced systems and logical design of large-scale digital computers.

Mr. Chao is a member of the AIEE and Eta Kappa Nu.

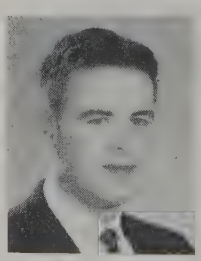
A. B. Clymer

Arthur Ben Clymer (M'57) was born in Cleveland, Ohio, on August 7, 1920. He received the B.A. degree in physics from Oberlin College, Oberlin, Ohio, in 1941, the M.S. degree in engineering-physics from The Ohio State University, Columbus, in 1946, and another M.S. degree in applied mathematics from the Massachusetts Institute of Technology, Cambridge, Mass. in 1948.

From 1942 to 1945 Mr. Clymer participated in the design of electromechanical special-purpose analog computers at Ford Instrument Company, Long Island City, N. Y., and between 1946 and 1948, he participated in the design of a missile flight simulator for the Dynamic Analysis and Control Laboratory at M.I.T. From 1949 to 1953, he served as the mathematical engineer for the Kimble Division of the Owens-Illinois Glass Company, Toledo, Ohio, employing theoretical analysis as an aid to product and process design, and from 1953 to 1955, he conducted theoretical and experimental research and served as research coordinator for Bituminous Coal Research, Inc., Columbus, Ohio. Since 1955 he has been engaged as a research specialist in the development and application of engineering computing methods for the Columbus Division of North American Aviation, Inc.

Mr. Clymer is a registered professional engineer in the state of Ohio. He is a member of the National Society of Professional Engineers, American Society of Mechanical Engineers, Society for Industrial and Applied Mathematics, Association for Computing Machinery, Industrial Mathematics Society, American Association for the Advancement of Science, Society for the Advancement of Management, Institute of Management Sciences, Pi Mu Epsilon, and Sigma Pi Sigma.

❖



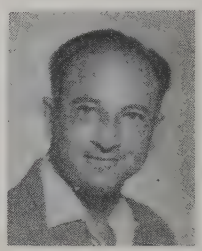
D. H. Daggett

Dan H. Daggett (M'57) was born in Fort Worth, Tex., on January 28, 1931. In 1953, he received the B.A. and B.S. degrees in electrical engineering from Rice Institute, Houston, Tex. After serving two years as an officer in the Civil Engineering Corps of the U. S. Navy, Mr. Daggett enrolled as a graduate student at Massachusetts Institute of Technology, Cambridge, Mass., which awarded him the S.M. degree in electrical engineering in 1956.

He then joined Convair, a Division of General Dynamics Corporation, Fort Worth, Tex., where he holds the title of senior aerophysics engineer in the airborne digital computer group. In this position, he has participated in digital systems work and the logical design of CORDIC. He is also a lecturer in electrical engineering at Southern Methodist University, Dallas, Tex.

Mr. Daggett is a member of Tau Beta Pi and Sigma Xi.

❖



S. P. Frankel

Stanley P. Frankel (M'55) was born in Los Angeles, Calif., on June 6, 1919. He received the B.A. degree in 1958 and the Ph.D. degree in 1942, both in physics, from the University of California at Berkeley.

Since 1943 he has made use of a number of digital computers for the solution of various industrial and scientific problems. He has served General Electric and other companies as a consulting engineer, primarily in logic design. He is the primary author of the logic designs of LGP-30, CONAC (a drum computer built by Continental Oil Company), the General Electric 200 and 210, M'AC, an academic design, NIC-NAC, a desk calculator, and a microwave computer.

Dr. Frankel is a member of the American Physical Society and the Association for Computing Machinery.

❖

E. O. Gilbert

Edward O. Gilbert was born March 29, 1930, in Joliet, Ill. He received the B.S. and M.S. degrees in electrical engineering, in 1952 and 1953, respectively, from the University of Michigan, Ann Arbor. In 1957, he received the Ph.D. degree in instrumentation engineering, also from the University of Michigan.

Since 1953, Dr. Gilbert has taught in the department of aeronautical engineering at the University of Michigan. He is presently an assistant professor there and his teaching activities are primarily in the instrumentation engineering degree program. His major fields of interest are automatic control, electronic analog computation, and flight simulation. He is now on a year's leave of absence as a technical staff member of Space Technology Laboratories, Inc., Los Angeles, Calif.

Dr. Gilbert is a member of Eta Kappa Nu, Tau Beta Pi, Phi Kappa Phi, and Sigma Xi.

Seymour Ginsburg (M'57) was born on December 12, 1927, in Brooklyn, N. Y. He received the B.S. degree in mathematics from the College of the City of New York in 1948, and the M.S. and Ph.D. degrees in mathematics from the University of Michigan, Ann Arbor, Mich., in 1949 and 1952, respectively.

S. Ginsburg

He was assistant professor of mathematics at the University of Miami, Coral Gables, Fla., from 1951 to 1955 and was a research analyst for Northrop Aircraft, Inc., Hawthorne, Calif., from 1955 to 1956, where he worked on problems arising in connection with the design and performance of digital computers. From 1956–1959 he was senior research engineer for the National Cash Register Company, Hawthorne, Calif., where he worked on applications of logic for use in digital computer systems, the theory of abstract machines, and the development of new methods of synthesis, analysis, and reduction of superfluous components. Since July, 1959 he has been head of the Systems Synthesis and Organization Section of the Data Processing Department in the Research Laboratories of Hughes Aircraft Co., Los Angeles, Calif.

Dr. Ginsburg is a member of the American Mathematical Society and the Association for Computing Machinery.

❖

Robert D. Gold (A'54) was born on June 21, 1931, in Brooklyn, N. Y. He received the B.S. degree from the College of the City of New York in 1953 and the M.S. degree from Cornell University in 1957, both in electrical engineering.

R. D. Gold

He spent three months with the RCA-Victor Division before entering the U. S. Army. Assigned to the Electronic Warfare Department under the Scientific and Professional Personnel Program, he worked on feasibility studies of radar countermeasure systems. He joined the RCA Laboratories in 1957, where he has worked on television circuits and kinescopes, and in the field of semiconductor devices. He has also been a lecturer in electrical engineering at the College of the City of New York.

Mr. Gold is a member of Tau Beta Pi and Eta Kappa Nu.

❖

Harry J. Gray, Jr. (S'45–A'46–M'55) was born in St. Louis, Mo., on June 24, 1924. He received the B.S. degree in electrical engineering in 1944, the M.S. degree in electrical engineering in 1947, and the Ph.D. degree in 1953, all from the University of Pennsylvania, Philadelphia.

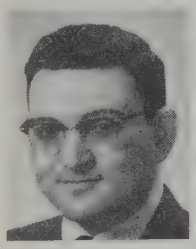
In 1946, after serving in the U. S. Navy as a radio specialist officer, he returned to the University of Pennsylvania. He worked on both the EDVAC and MSAC computers and then was associated with the development of a digital operational flight trainer (UDOFT). He contributed to the basic system organization of UDOFT, introduced the stability chart used to estimate the effect of accumulated truncation errors in the numerical integration of the equations of the aircraft motion, and contributed to the design of the logical packages of UDOFT, and to the peripheral equipment. In 1954, he joined the Remington Rand Univac Division of the Sperry-Rand Corporation, Philadelphia, Pa., where he worked on magnetic and transistor circuits. He developed the high-speed circuit system of LARC, and when he left in 1957, he held the position of staff engineer. At present he is associate professor of electrical engineering in the Moore School of the University of Pennsylvania, is continuing development of the Multiple-Cockpit Trainer, and is a consultant for government and industry.

H. J. Gray, Jr.

Dr. Gray is a member of the Association for Computing Machinery, Sigma Xi, Tau Beta Pi, and Eta Kappa Nu.

❖

Jack Hilibrand (S'50–M'57) was born in New York, N. Y., on September 15, 1930. He received the B.S. degree in electrical engineering from the College of the City of New York in 1951. At M.I.T. he was a National Science Foundation Fellow for two years and a research assistant at the Research Laboratory of Electronics for two years. In 1956 he received the Sc.D. degree from M.I.T. for work in the analysis of excess physical noises.



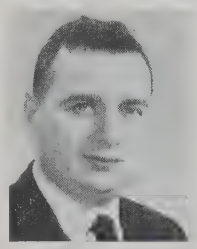
J. Hilibrand

Since graduation, he has been a member of the technical staff at the RCA Laboratories in Princeton, N. J., where he is now engaged in semiconductor device design and analysis. He also holds the rank of lecturer in the Graduate Division of the School of Technology at the College of the City of New York.

Dr. Hilibrand is a member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi.

❖

Peter J. Isaacs (M'55) was born on May 27, 1926, in New York, N. Y. He received the B.S. degree in physics from Rensselaer Polytechnic Institute, Troy, N. Y., in 1946, and Columbia University, New York, N. Y., awarded him the M.A. and Ph.D. degrees in physics in 1947 and 1951, respectively.



P. J. Isaacs

He served in the U.S. Navy from 1944 to 1946 and then was employed by the Government Contract Division of Columbia University, New York, N. Y. In 1953, he joined the Sperry Gyroscope Company, Great Neck, N. Y., where he is now engineering section head for digital components research in the Surface Armament Division. His experience with Sperry has been principally in the field of digital computer development.

Dr. Isaacs has one patent application pending and is a member of the American Physical Society.

❖

William L. Kilmer (A'54) was born in Easton, Pa., on July 5, 1932. He received the B.S. and M.S. degrees in electrical engineering from Pennsylvania State University, University Park, in 1954 and 1955, respectively, and the Ph.D. degree in dielectrical engineering from the University of Michigan, Ann Arbor, Mich., in 1958.

W. L. Kilmer

Nearly all of his work has been in the general area of digital computer design. Through the fall of 1958, he was successively employed at the Research and Development Center, Griffiss Air Force Base, Rome, N. Y.; Bell Telphone Laboratories, Inc., Whippany, N. J., where he participated in the logical design of the Leprechaun Computer; The University of Michigan Engineering Research Institute, where he worked on high-speed computer components; and IBM Research, Yorktown Heights, N. Y.

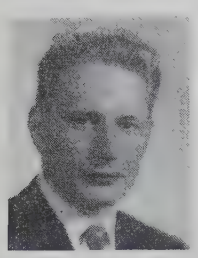
Since October, 1958, he has held an assistant professorship in electrical engineering at Montana State College, Bozeman, and a research appointment at the Montana State College Electronics Research Laboratory, where he is presently heading a computer research group to work in the areas of logical design procedure, and abstract automata.

Dr. Kilmer is a member of the ACM, Sigma Xi, and several engineering honor societies.

❖

Walter F. Kosonocky (S'54–M'55) was born in Sieradz, Poland, on December 15, 1931. He was awarded the B.S. and M.S. degrees in electrical engineering by Newark College of Engineering, Newark, N. J., in 1955 and 1957, respectively. Since 1957, he has been studying towards the Sc.D.

degree in engineering at Columbia University, New York, N. Y.; for this program he was awarded the David Sarnoff Fellowship for the academic year 1958–1959.



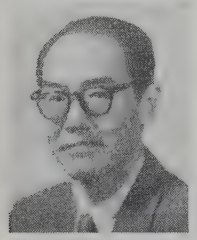
W. F. Kosonocky

He was a lecturer at Newark College of Engineering during 1958–1959. Since 1955 he has been employed at the Radio Corporation of America Laboratories in Princeton, N. J., where he has been engaged in computer components and systems research.

Mr. Kosonocky is a member of Tau Beta Pi and Eta Kappa Nu.

❖

Arthur W. Lo (S'43–A'50–SM'56) was born in Shanghai, China, on May 21, 1916. He received the B.S. degree in physics from Yenching University, China, in 1938, the M.S. degree in physics from Oberlin College, Oberlin, Ohio, in 1946, and the Ph.D. degree in electrical engineering from the University of Illinois, Urbana, Ill., in 1949.



A. W. Lo

He taught physics and electronics in several colleges, both in this country and abroad, before joining the Radio Corporation of America Laboratories, Princeton N. J. in 1951. His major interest has been in solid-state devices for digital systems including transistors, magnetic devices, and the transfluxor, and in millimicrosecond digital components and techniques. He is presently leading a research group at RCA which is working on the application of solid-state switching and storage devices. He has contributed a number of technical papers, lectures, and United States and foreign patents and co-authored a book, "Transistor Electronics."

Dr. Lo is a member of Sigma Xi, Phi Kappa Phi, Pi Mu Epsilon, and Eta Kappa Nu.

❖

Ralph E. Meagher (M'49) received the B.S. degree from the University of Chicago in 1938, the M.S. degree from Massachusetts Institute of Technology in 1939, and the Ph.D. degree in physics from the University of Illinois in 1949. He was a pre-doctoral National Research Fellow during 1945–1948.

R. E. Meagher

During the war he was a staff member and associate group leader in the Radiation Laboratory of M.I.T. There he carried out work in the design and development of search radar, and was particularly associated with the design of the indicator systems for radar used by the Navy. In 1948 he was awarded the Presidential Certificate of Merit for his work. From 1949 to 1952, he was associated with the design of the ORDVAC and the ILLIAC at the University of Illinois, serving as chief engineer. He is now there as research professor of physics and electrical engineering and head of the Digital Computer Laboratory, where research is currently under way on new computer techniques and applied mathematics.

Dr. Meagher is a fellow of the American Physical Society, and is a member of the Association for Computing Machinery. He served for two years as Editor of these Transactions.

❖

Charles W. Mueller (S'55–A'36–SM'45–F'49) was born in New Athens, Ill., on February 12, 1912. He received the B.S. degree in electrical engineering from the University of Notre Dame in 1934. He was awarded the S.M. degree in electrical engineering in 1936 and the Sc.D. in 1942, both from the Massachusetts Institute of Technology.

C. W. Mueller

From 1936 to 1938 Dr. Mueller was associated with Raytheon Production Corporation, first in the engineering supervision of factory production of receiving tubes, and then in the development of gas-tube voltage regulators and cold-cathode thyratrons. In the fall of 1938 he returned to the M.I.T. where he worked on a government contract on the development of gas-filled special-purpose tubes for counting operations. From 1942 to the present he has been a member of the RCA Laboratories Division, where he has been engaged in research on high-frequency receiving tubes, secondary electron emission phenomena, and solid-state devices.

Dr. Mueller a member of the American Physical Society and Sigma Xi.

❖

Saburo Muroga was born in Numazu, Japan, on March 15, 1925. He received a degree in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1947. He also studied at the Research Laboratory of Electronics of the Massachusetts Institute of Technology, Cambridge, Mass., in 1953, and at the Digital Computer Laboratory of the University of Illinois, Urbana, in 1954.

S. Muroga

He was engaged in theoretical research on various pulse modulation methods and the narrow-band voice transmission systems in the Railway Technical Laboratories from 1947 to 1950, and in the Radio Regulatory Commission from 1950 to 1951. In 1951, he joined the staff of the Electrical Communication Laboratories of the Nippon Telegraph and Telephone Public Corporation, Tokyo, Japan, and since then has been engaged in research on the information theory. He is in charge of construction of the parametron digital computer, MUSASINO-1. His recent interest is an algebraic theory of logical elements on majority decision principle.

Dr. Muroga is a member of the Institute of Electrical Communication Engineers of Japan and of the Physical Society of Japan.

❖

Douglas B. Netherwood (M'56–SM'59) was born in Dallas, Tex., on May 8, 1920. He received the B.S. degree from the U. S. Military Academy at West Point, in 1943, the M.S. degree from the Air Force Institute of Technology, Dayton, Ohio, in 1957, and is now a doctoral applicant at the University of Michigan, Ann Arbor.

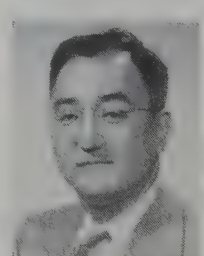
D. B. Netherwood

He served in India and Burma during World War II and was advisor on research and development to the Nationalist Government of China from 1946 to 1947. He was a communications and electronics officer in the Strategic Air Command from 1947 to 1955. During the past year he has been chief of the Advanced Development Branch, Electronic Components Laboratory, Wright Air Development Center, Dayton, Ohio. His field of specialization is the theory of machines and automata.

Major Netherwood is a member of the Association for Computing Machinery and the Association for Symbolic Logic.

❖

Hiroshi H. Nishino (S'41–A'43–SM'54) was born on June 25, 1920, in Tsuruga, Japan. He received the B.S. degree in electrical engineering in 1946 and the E.E. degree in 1948, both from Stanford University, Stanford, Calif.



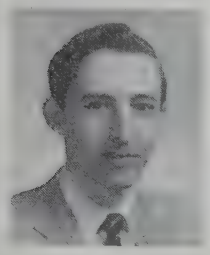
H. H. Nishino

He served as a senior research engineer at the Dalmo Victor Co., San Carlos, Calif., where he was engaged in microwave techniques and airborne antenna design. He has been with the Radio Corporation of America since 1950 except for a period of a year and a half (1956–1958) when he was studying and doing research at the Moore School of Electrical Engineering,

University of Pennsylvania, Philadelphia. His work at RCA started with the design efforts on the multiple TV and FM broadcasting antenna for the Empire State Building, and later he worked on fire-control, navigation, and scheduling-control computers for military applications. He is currently in the Systems Engineering activity of the Missile and Surface Radar Division of RCA, Moorestown, N. J.

❖

Lubomyr S. Onyshkevych (S'54–M'56) was born in Lviv, Ukraine, on February 3, 1933. He was awarded the B.S. degree in electrical engineering by the College of the City of New York, N. Y., in 1955. In 1957, he received the M.S. degree in electrical engineering from Massachusetts Institute of Technology, Cambridge, Mass., and is at present working toward a Sc.D. degree there.



L. S. Onyshkevych

From 1955 to 1957, he worked as a research assistant at the Research Laboratory of Electronics at M.I.T. on transfluxers and other magnetic computer components. He joined the Radio Corporation of American Laboratories in 1957, and as a member of the technical staff at the David Sarnoff Research Center, he worked in computer components, including magnetic and parametric devices. He returned to M.I.T. in September, 1959, where he is currently a research assistant.

Mr. Onyshkevych is a member of Tau Beta Pi and Eta Kappa Nu, and an associate member of Sigma Xi and the American Institute of Electrical Engineers.

❖

William C. G. Ortel was born on September 28, 1926, in Spokane, Wash. He received the B.S. degree in 1949, the M.S. degree in 1950, and the Ph.D. degree in 1953, all in physics from Yale University, New Haven, Conn.

W. C. G. Ortel

He held the National Science Foundation postdoctoral fellowship in 1954–1955, doing cosmic-ray research at the Institute for Theoretical Physics, Copenhagen, Denmark. Since 1956 he has been at the Bell Telephone Laboratories, Murray Hill, N. J., where he is doing research on high-speed digital computers.

❖

Marvin C. Paull was born in New York, N. Y., on May 5, 1929. He served as an electronics technician in the United States Navy from 1946 to 1948. In 1952 he received the B.E.E. degree from the Clarkson College of Technology, Potsdam, N. Y.

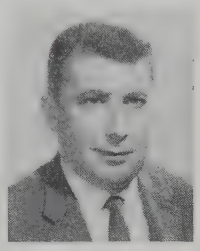
M. C. Paull

He was with the Arma Corporation for one year ending in 1953. Since then he has been a member of the technical staff of the Bell Telephone Laboratories, Whippany, N. J. In this capacity he has been concerned with magnetic core circuitry, logical design, sequential switching circuit theory, and programming research.

Mr. Paull is a member of Eta Kappa Nu and Tau Beta Pi.

❖

William Sauter was born on October 15, 1927, in Nanuet, N. Y. He received the B.S. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, N. Y., in 1951.



W. Sauter

He served in the U. S. Navy from 1946 to 1947 and in March, 1951 joined the Sperry Gyroscope Company, Great Neck, N. Y., where he is now a senior engineer in the Digital Systems Department of the Surface Armament Division. Since his association with Sperry, he has been principally concerned with analog and digital computers. His most recent experience involved feasibility studies of microwave computer components in connection with ultra high-speed computers.

❖

Fred Sterzer (M'56) was born in Vienna, Austria, on November 18, 1929. He received the B.S. degree in physics in 1951 from the College of the City of New York and the M.S. and Ph.D. degrees in 1952 and 1955, respectively, from New York University.

F. Sterzer

From 1952 to 1953 he was employed by the Allied Control Corporation, in New York, N. Y. From 1953 to 1954 he was an instructor in physics at the Newark College of Engineering, Newark, N. J. and a research assistant at New York University. He joined the RCA Tube Division in Harrison, N. J., in 1954, and transferred to the Princeton, N. J. branch in 1956, where he is now group leader in microwave physics. His work has been in the fields of microwave spectroscopy, traveling-wave tubes, backward-wave oscillators, solid-state microwave amplifiers and oscillators, and microwave computing circuits.

Dr. Sterzer is a member of Phi Beta Kappa, Sigma Xi, and the American Physical Society.

❖

Carl F. Stocker was born on March 31, 1932 in Chicago, Ill. He received the Bachelor of Science degree in physics from Loyola University, Chicago in 1954. In 1956, he received M.S. degrees in physics and in chemistry from the University of Illinois, where he did research on the surface properties of polymers.

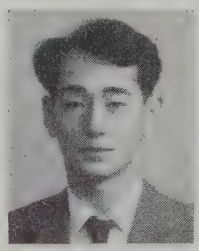
C. F. Stocker

He taught physics at Loyola University before joining the Electrical Design Group at the Hawthorne plant of Western Electric and since 1956 has been associated with RCA Laboratories, where his research on phosphors led to an Achievement Award. He is presently concerned with research on semiconductors and device designs, particularly variable capacitance and avalanche diodes.

Mr. Stocker is a member of the Electrochemical Society.

❖

Kensuke Takashima was born February 23, 1928, in Fukuoka, Japan. He received a degree in physics from the University of Tokyo, Tokyo, Japan, in 1950.



K. Takashima

After his graduation he joined the staff of the Electrical Communication Laboratories, Nippon Telegraph and Telephone Public Corporation. At that time he was engaged in research on the electron dynamics of microwave tubes. In 1951, he was transferred to the transmission research department, where he was engaged in the practical application of the information theory and the design and development of the electronic analog correlator. Since 1954, his work has been in the field of electronic digital computers. He has taken part in the basic research on the parametron, and has worked on the design and construction of the parametron digital computer MUSASINO-1.

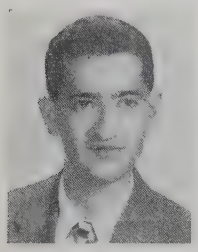
He is currently studying high-speed electronic logical components.

Mr. Takashima is a member of the Institute of Electrical Communication Engineers of Japan and of the Physical Society of Japan.

❖

Stephen H. Unger was born in New York, N. Y., on July 7, 1931. In 1952 he received the B.E.E. degree from the Polytechnic In-

stitute of Brooklyn, N. Y., and in 1953 and 1957 he received the S.M. and Sc.D. degrees, respectively, in electrical engineering from the Massachusetts Institute of Technology, Cambridge, Mass.



S. H. UNGER

From 1955 to 1957 he was a research assistant at the Research Laboratory of electronics at M.I.T., where he investigated various aspects of sequential switching circuit theory. Between 1952 and 1955 he held summer positions with the General Electric Company, International Business Machines Corporation, and the Bell Telephone Laboratories. Since 1957, he has been a member of the technical staff of the Bell Telephone Laboratories at Whippany, N. J., where he is engaged in digital systems research, and telephone systems development.

Dr. Unger is a member of Sigma Xi, Eta Kappa Nu, and Tau Beta Pi.

Alvin L. Vivatson was born on April 26, 1929, in Svold, N. Dak. He received the B.A. degree in mathematics in 1951 from State Teachers College, Valley City, N. Dak.

A. L. VIVATSON

In 1952 he joined for a six-month training program with RCA-Victor, and during the next four years was in business for himself. In 1956 he joined the staff of the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, as a research assistant. Since then he has worked with numerical analysis and programming for the UDOFT (a digital operational flight trainer) and with the system design and the logical design for a multi-cockpit trainer computer.

Jack E. Volder (A'55–SM'58) was born in Fort Worth, Tex. on September 9, 1924. He received the B.S.E.E. degree from Texas Technological College, in 1949.

From 1943 to 1946 he served with the USAAF as a B-24 flight engineer. From 1949 to 1951, he was employed in the Control Engineering Department of the Allis Chalmers Manufacturing Company, Milwaukee, Wis. In 1951 he joined the field service department of Convair, Fort Worth, Tex., where he designed test and ground support equipment; in 1955, he transferred into the aerophysics department where he initially worked in the field of analog-digital conversion. Later his investigations into the application of existing digital computing techniques to specific airborne computing requirements led to his conception of the computing algorithm of the CORDIC computer.

J. E. VOLDER

Mr. Volder is a member of the AIEE, the National Society of Professional Engineers, and Tau Beta Pi.

Reviews of Books and Papers in the Computer Field

In this issue, IRE TRANSACTIONS ON ELECTRONIC COMPUTERS inaugurates a new feature —regular reviews of books and significant articles in the computer field. The reviews are arranged for and compiled by E. J. McCluskey, Jr., Associate Professor of Electrical Engineering at Princeton University, formerly with Bell Telephone Laboratories. If you are interested in assisting in this effort, contact him. In any event, please let us know your reaction to this new feature; all comments welcome. —*The Editor.*

Finite Automata and Their Decision Problems—M. O. Rabin and D. Scott. (*IBM J. Res. & Dev.*, vol. 3, pp. 114–125; April, 1959.)

One of the significant problems of continuing interest in research in automata is to find mathematical models which are capable of the most sophisticated behavior possible without being hopelessly unrelated to reality. Motivated by this aim, the authors of this paper consider a number of models of sequential machines which require only a finite number of internal states and a finite, but variable-length, tape for their operation. They are able to characterize the action of some of these models and show that some basic questions about them are not answerable.

The first model considered is designated a one-way, one-tape automaton. Roughly speaking, this is a finite-state machine with a single output and an input consisting of a finite list of symbols taken one at a time from a finite alphabet. After the last input symbol is read, the output is determined as either 0 or 1; *i.e.*, the tape is "rejected" or "accepted." This model is obviously equivalent to ones studied by many other workers, *e.g.*, Moore,[1] so that most of the results of this section are not new. They obtain very simple characterizations of sets of acceptable tapes, *i.e.*, those sets of tapes for which there is a one-way, one-tape automaton which accepts exactly those tapes in these and rejects all others. These characterizations lead to simple proofs of a number of interesting but unsurprising theorems.

The first generalization the authors consider is to a two-way, one-tape automaton. As its name implies, this automaton differs from the previous one in that the input tape is allowed to move backward as well as forward. A tape is accepted if it ends up at the last symbol in a state which has an output of one, without, of course, going off the end of the tape previously. The surprising theorem obtained is that, for every two-way automaton, there exists a one-way automaton which accepts exactly the same class of tapes. This theorem is proved in a beautifully simple manner in an accompanying article by Shepherdson.[2]

The next generalization considered is about multitape automata for which the machine reads inputs from one tape at a time, the particular tape to be read being determined by the state of the machine. Here again we have the possibility of one-way or two-way motion of the tapes; and it is shown that two-tape, one-way automata are closely related to one-tape, one-way automata. However, it turns out to be very different for two-way machines; and, in fact, it is shown that there are sets of tapes definable by two-tape, two-way automata which are not definable by any two-tape, one-way automaton.

One of the most interesting concepts introduced by the authors is that of a nondeterministic automaton. Although they are interested in it as a tool for simplifying the proofs of many of the theorems, it conceivably could have applications in the problem of synthesis of sequential machines. A nondeterministic finite automaton A over the alphabet Σ is a system $A=(S, M, S_0, F)$ where S is a finite set, M is a function on $S\times\Sigma$ with values in the set of all subsets of S, and S_0 and F are subsets of S. Here S is the set of internal states of the machine, S_0 is the set of states in which the machine may begin operation, and F is the set of states with output one. Since M has values in the set of all subsets of S (rather than S itself), an internal state and an input symbol determine a set of possible successor states rather than exactly one successor state. A tape $x=\sigma_0, \sigma_1, \sigma_2, \cdots, \sigma_n$, and is accepted by the automaton A if and only if there is a sequence of states $s_0, s_1, \cdots, s_n$ such that

$$
\begin{aligned}
&1)\ s_i \in S && i = 0, \cdots n\\
&2)\ s_0 \in S_0 &&\\
&3)\ s_i \in M(S_{i-1}, \tau_{i-1}) && i = 1, \cdots n\\
&4)\ s_n \in F. &&
\end{aligned}
$$

Thus, a nondeterministic automaton is not a probabilistic machine but is one in which the machine may have many choices at various stages of its operation, and a tape is accepted if there is at least one choice at every stage which leads to the tape's acceptance.

Although a nondeterministic automaton appears to be more general than an ordinary one-tape, one-way automaton, this turns out not to be the case, and the authors show that for every nondeterministic automaton, there is an ordinary one which accepts exactly the same class of tapes. This theorem enables the authors to carry out many proofs in terms of nondeterministic automata rather than the corresponding ordinary ones. Since there are usually fewer internal states required in such a machine, this tends to simplify the proofs. For the same reason, these machines might form a useful intermediate step in the synthesis of a sequential machine to carry out a specified job.

T. H. CROWLEY
Bell Telephone Labs.
Murray Hill, N. J.

A Self-Reproducing Analogue—L. S. Penrose and R. Penrose. (*Nature*, vol. 179, pp. 1183; June 8, 1957.)
Mechanics of Self-Reproduction—L. S. Penrose. (*Ann. Human Genet.*, vol. 23, pt. I, pp. 59–72; November, 1958.)
Automatic Mechanical Self-Reproduction,—L. S. Penrose. ("New Biology," Penguin Books, Harmondsworth, England, no. 28, pp. 92–117; January, 1959.)
Self-Reproducing Machines—L. S. Penrose. (*Sci. Amer.*, vol. 200, pp. 105–114, 202; June, 1959.)

These papers describe several surprisingly simple and ingenious solution to von Neumann's problem of making self-reproducing automata. All of the logic and the programming by which the assembling of the parts in the correct arrangement and in the correct states is controlled, is done by mechanical means, using hooks and latches which depend on gravity and friction only. No electrical, magnetic, or chemical forces are used. The paper from *Nature* is a short preliminary announcement, superseded by the other papers. Each of the three later papers contains material which supplements the others, such as supplementary explanations of ideas which are glossed over briefly in the other papers. L. S. Penrose, a British geneticist, has designed the various models to have many analogies to the chemical and biological properties of living matter. There are a small number of different kinds of parts present, analogous to the

[1] E. F. Moore, "Gedanken-experiments on sequential machines," in "Automata Studies," Princeton University, Princeton, N. J., pp. 129–153; 1956.
[2] J. C. Shepherdson, "The reduction of two-way automata to one-way automata," *IBM J. Res. & Dev.*, vol. 3, pp. 198–199; April, 1959.

molecules of the organic compounds from which living organisms are formed. These parts are arranged at random on a horizontal surface, and the energy necessary for their operation is provided by mechanical vibration of this surface, producing motions analogous to the thermal motions of molecules. Each of the parts has different states, which have different amounts of potential energy. If a complete machine (called a seed) made of these parts is present, it causes the parts to arrange themselves into copies of the seed. If no seed is present, no "spontaneous generation" occurs. In even the simplest of the models, there are two different possible seeds which can reproduce themselves. In the more advanced models, a seed can contain arbitrarily long chains of information storing units, corresponding to the chains of molecules in a chromosome. Several of the models described have been actually constructed and successfully operated.

The author notes that his ideas of self-reproduction by entanglement of interlocking shapes were anticipated by Lucretius, who wrote more than 2000 years ago. But Penrose's step forward in the design and simplification of these models is a major one. The construction of further models of this kind provides many interesting problems in logical and mechanical design, which may contribute ideas to theoretical biology, but which will certainly enlarge our knowledge of the capabilities and limitations of machines.

EDWARD F. MOORE
Bell Telephone Labs.
Murray Hill, N. J.

Reliable Stochastic Sequential Switching Circuits—A. A. Mullin. [*Commun. & Electronics*, no. 39 (*Trans. AIEE*, pt. 1, vol. 77), pp. 606–612; November, 1958.]

This paper is concerned with sequential switching circuits using imperfect relays as components. The model used for the imperfect relays is the "crummy" relay discussed by Moore and Shannon.[1] In this model, whenever a change occurs in the excitation of a relay, its various contacts, acting independently of one another, malfunction with some nonzero probability.

In a straightforward manner, Mullin shows how to get an upper bound on the probability of an error occurring during any transition between stable states of a given sequential circuit built with crummy relays. His procedure takes into account the effects of combinational (or static) hazards as discussed, by Huffman.[2]

A three-part synthesis procedure is suggested for obtaining reliable sequential switching circuits. The first part consists of using the "classical" synthesis procedure developed by Huffman[3] to obtain minimal circuits. The second part consists of an analysis (as indicated in the preceding paragraph) to determine error probabilities for the various transitions, based on the characteristics of the given crummy relays. Finally, the Moore-Shannon procedure for replacing each contact with an array of four or more contacts from the same relay is used where necessary to attain the desired degree of reliability.

The paper is well organized and readable, but it consists essentially of a straightforward application of the crummy relay concept to sequential switching circuits.

S. H. UNGER
Bell Telephone Labs.
Whippany, N. J.

[1] E. F. Moore and C. E. Shannon, "Reliable circuits using less reliable relays," *J. Franklin Inst.*, vol. 262, pp. 191–208, 281–297, September/October, 1956.
[2] D. A. Huffman, "The design and use of hazard-free switching networks," *J. Assoc. Comp. Mach.*, vol. 4, pp. 47–62; January, 1957.
[3] D. A. Huffman, "The synthesis of sequential switching circuits," *J. Franklin Inst.*, vol. 257, pp. 161–190, 275–303; March/April, 1954.

Analog Simulation—W. J. Karplus. (McGraw-Hill Book Co., Inc., New York, N. Y.; 1958. 426 pp.+4 index pp.)

Because of the widespread use of digital and analog computers, there has been some lessening of interest in the field of direct simulation. For this reason, Professor Karplus' book is especially welcome, and will surely revive some deserved enthusiasm in an area which still offers many interesting possibilities. The book should be useful both as a textbook for a college course at about the junior or senior level and as a convenient source.

The volume starts with an introduction that discusses the basic ideas of duals and analogs, and discusses such general matters as convenience of electrical analogs and the "controversy" between analog and digital methods. The introduction ends with several conclusions which may elicit some disagreement among its readers.

Chapters two, three, and four constitute the first of three parts into which the book is divided. These excellent chapters concern the physical and mathematical principles which underlie the partial differential equations of physics. The point of view is particularly well suited to the electrical engineer. Transformations such as conformal mapping are presented and summarized. Finite difference approximations and their associated errors are described in considerable detail, and all material is drawn together in excellent tables and summaries.

Part Two is entitled "Analog-Simulation Systems," and reviews most of the techniques based on conducting sheets, electrolytic tanks, and other miscellaneous methods. The important topic of resistance networks receives a full chapter, including Liebmann's method for dealing with the diffusion equation without capacitive elements and similar techniques for the biharmonic and wave equations. The following chapter describes the techniques built around the use of RC, LC, and RLC elements.

The briefest possible discussion of analog computers follows. The emphasis is directed toward the use of operational amplifiers in conjunction with passive simulation networks. This kind of application, however, makes heavy demands on these amplifiers. Their satisfactory use may require a somewhat deeper background in electronic circuits than is assumed here. There follows a chapter, included for completeness, on nonelectric analogs such as membranes, fluid mappers, and Rayleigh's pin-wire models of Laplacian field problems.

The last part of the book consists of four chapters which effectively summarize the preceding material in terms of applications. The first of these concerns equations of the elliptic type, and takes its examples from fields such as equipotentials in an electron gun, a high-voltage insulator, and a magnetron; some heat transfer problems, viscous and nonviscous flow, and several others. The second chapter considers applications in parabolic equations inluding some very difficult problems in nonlinear heat conduction, and compressible fluid flow. The last two chapters discuss applications of the techniques to hyperbolic equations and biharmonic equations, respectively. Examples for the former include the vibrations of elastic strings and membranes, Schroedinger's equation, and Maxwell's equations, particularly in cavity and waveguide problems. The latter chapter solves beam and plane stress problems.

The book concludes with an excellent bibliography organized by dates as well as by fields. The topic of elasticity, for example, is divided into five sections starting with the principal papers between 1900 and 1930, then 1931 to 1935, and concluding with the period 1951 through 1956. Similar breakdowns are made in the remaining areas. All in all the book does a commendable job of organizing a large amount of detail and presenting it in a readable form. Particularly impressive are its many fine illustrations, its tables, and its chapter summaries. The editors as well as the author have contributed to making this a thoroughly useful and readable book.

I. B. PYNE
Princeton University
Princeton, N. J.

Abstracts of Current Computer Literature

(THROUGH APRIL, 1959)

This issue of the TRANSACTIONS continues the literature abstracting service recently inaugurated by the PGEC. The abstracts and associated subject and author indexes were prepared on a commercial basis by a Massachusetts firm under management of Dr. Geoffrey Knight, Jr. This firm is best known for publication of the abstract journal "Semiconductor Electronics." Local volunteer support in monitoring this endeavor has been furnished by Messrs. P. R. Bagley and R. P. Mayer of The Mitre Corporation, and F. E. Heart of Lincoln Laboratory, M.I.T.

Comments on this abstract service are welcomed. Please let us know if it is useful to you, or if it could be improved. Additional copies of these abstracts are available from IRE Headquarters, 1 East 79th St., New York 21, N. Y., at $1.00 per copy, or $3.50 for the set of four published in 1959. —*The Editor.*

A-1: EQUIPMENT—THEORETICAL DESIGN

261
Symbolic Design Techniques Applied to a Generalized Computer, by I. S. Reed (MIT Lincoln Lab.); *U. S. Gov. Res. Repts.*, vol. 31, p. 240(A), April 17, 1959; PB 130 877.

Techniques for the symbolic design of digital computers are developed and discussed. These techniques are then applied to the development of a generalized computer which has an internal nature that can be programmed with a sequence of micro-instructions. This machine has the capability of simulating the internal nature of any other computer that utilizes the same type of main memory.

262
Logical Design Methods, by R. K. Richards; *Proc. WJCC*, pp. 179–181; May 6–8, 1958.

The advantages of computer design by means of block diagrams, rather than by means of logical equations, are discussed.

263
Machine Language In Digital Computer Design, by H. L. Engel (Ramo-Wooldridge Corp.); *Proc. WJCC*, pp. 182–186; May 6–8, 1958.

If the description of the logical structure of a digital computer is in a form that can be processed by a digital computer, this description is said to be in a machine language. The advantages of machine language in the design of digital computers are discussed. Machine language can be used to produce logic and usage tabulations, to generate a wiring tabulation which indicates all signal wiring in the computer, and to simulate the logic of one computer by a completely different computer. The logic and usage tabulations facilitate the logical and production design of the computer, permit easy modifications, and make computer maintenance simple. The wiring tabulation can be produced faster and cheaper than a wiring diagram and can be displayed in a simpler form. Simulation of computer logic results in considerable savings of time and money in the design, construction, and check-out of a new computer. Logic simulation makes maintenance easier since the behavior of a computer with a faulty diode or flip-flop can be simulated. Lists which describe the behavior of the computer when a particular component fails can then be made.

264
The Design of Conditional Probability Computers, by A. M. Uttley (Nat. Physical Lab., England); *Infor. and Control*, vol. 2, pp. 1–24; April, 1959.

A special-purpose computer which calculates conditional probabilities is described. The input to the computer is a set of channels which are in either an active on an inactive state. At any instant a particular set of channels will, in general, be active; the computer calculates the conditional probability of all the other channels, based on what has happened in the past. The computer can be extended to forecast the probability of future signals, and the past can be weighed in any desired manner. Such a computer uses the alogical principle of induction, and it can imitate many forms of animal learning. Full details for the construction of such machines are given.

265
Time Multiplexing as Applied to Analog Computation, by E. Rawdin (RCA); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 42–47; March, 1959.

A multiplexer that can perform a common dynamic operation upon several sets of inputs utilizing equipment for one dynamic operation is described. The result of the multiplexing will yield the corresponding several sets of outputs as though each set of inputs were operated upon separately by the dynamic operation. This device is useful when the dynamic operation to be shared involves relatively expensive equipment such as electro-mechanical gear and/or electronic computing or measuring circuits. It is particularly useful to reduce the number of components required when implementing problems in a simulation laboratory.

266
Unifying Design Principle for the Resistance Network Analogue, by F. C. Gair (Associated Electrical Industries Ltd.); *Brit. J. Appl. Phys.*, vol. 10, pp. 166–172; April, 1959.

The simplification of the design of resistance network analogs is discussed. Simplification is achieved, without any loss of accuracy, by the use of the "cell principle," which is an extension and generalization of the method of MacNeal. Poisson's equation is integrated over the volume of a representative small cell. In this form it is much easier to appreciate the analogy with the resistance network, and therefore to arrive immediately at the relevant design parameters. The method applies not only to the usual type of network with equal meshes, but also to networks with unequal meshes, or subdivided meshes, and further to the representation of the Dirichlet, Hermann, or Fourier boundary conditions on rectangular or arbitrarily curved boundaries. More complicated equations than Poisson's can also be represented by a resistance network. By consideration of these equations as generalizations of Poisson's equation in an appropriate Riemann space, the corresponding design parameters are achieved. This formulation can be used equally well as the basis for any numerical solution of the equations.

A-2: EQUIPMENT—COMPONENTS AND CIRCUITS

267
Magnetics for Computers—a Survey of the State of the Art, by J. A. Rajchman (RCA Labs.); *RCA Rev.*, vol. 20, pp. 92–135; March, 1959.

Present-day applications of magnetics to random-access memories and logic switching are surveyed and appraised. The memories discussed include core arrays, external addressing, apertured plates, thin films, and twistors. The survey of switching includes combinatorial switches, shift registers, current steering, voltage drives, transistor coupling, parametron, transfluxors, transfluxor current steering, and multi-apertured transfluxors.

268
Recent Advances in Magnetic Devices for Computers, by D. H. Looney (Bell Tel. Labs., Inc.); *J. Appl. Phys.*, suppl. to vol. 30, pp. 38S–42S; April, 1959.

Some of the newer magnetic computer elements which may replace ferrite toroids in memory matrices are described. Ferrite materials are being used as sheets and multiaperture devices. The sheets show great promise in achieving a cost reduction. The multiaperture devices in memory applications offer the advantages of increased speeds, a wider operating temperature range, and nondestructive readout. The ferromagnetic materials are used in the form of thin films and wrapped wire. The films are capable of increased switching speeds at nominal current drives. The use of magnetic wire as a storage element offers new fabrication technology, a wider operating temperature range, and new functional structures. A comparison is made to highlight the device characteristics of the various structures.

269
Producing High-Performance Low-Cost Magnetic Memory Cores for An Expanding Digital Computer Market, by C. L. Snyder (Gen. Ceramics Corp.); *Computers and Automation*, vol. 8, pp. 9–11; March, 1959.

The history, production, testing, and performance of low-cost ferrite cores are described.

270
Study of the Residual States of Ferrite Cores in Computer Memory Operation, by W. M. Overn and V. J. Korkowski (Remington Rand Univac); *J. Appl. Phys.*, suppl. to vol. 30, pp. 52S–53S; April, 1959.

A study of the two complementary residual states of magnetic induction used to represent binary information in a ferrite-core storage system is discussed. The flux density found at the two states depends on any large fields which may have saturated the core in the past, as well as on the nature of the driving fields employed in the system. The partial-select noise amplitude is a function of the residual magnetic induction and saturation history. By applying this information, one can eliminate "delta" noise. The role of the postwrite disturb pulse in preventing errors under certain abnormal operating conditions has been determined.

271
Millimicrosecond Switching Properties of Ferrite Computer Elements, by W. L. Shevel, Jr. (IBM Res. Lab.); *J. Appl. Phys.*, suppl. to vol. 30, pp. 47S–48S; April, 1959.

An examination of the switching properties of square-loop ferrites is presented. Switching times have been studied over the range 5 μsec to 10 mμsec. The switching parameters, threshold field and switching constant, have been studied as a function of temperature and of ceramic processing. The plot of inverse of switching time vs applied fields displays three nearly linear portions for which the slopes vary by a factor of from two to ten. Therefore, the inverse slope known as the switching constant of the material has three values; this is interpreted as indicating three mechanisms to be responsi-

ble for the process of flux reversal, each mechanism being dominant over a certain region of the switching curve. These mechanisms are proposed as being wall motion, incoherent rotation, and coherent rotation. A model allowing a coherent rotation process is proposed. Data are presented for several ferrites which have widely varying properties.

272
Evaluation of New High-Speed Magnetic Ferrite System for Use in Computer Components, by B. R. Eichbaum (IBM Corp.) *J. Appl. Phys.*, suppl. to vol. 30, pp. 49S–52S; April, 1959.

A study of the $CdO \cdot MnO \cdot Fe_2O_3$ ferrite system is discussed. Some of the compositions of this new system exhibiting square hysteresis loops have switching constants as low as 0.200 oe μsec. Such materials, in comparison with those of the $MgO \cdot MnO \cdot Fe_2O_3$ system, have a much lower coercive force, require lower driving currents, and have a flux reversal or switching time which is five times as fast. These materials have been used in fast switching multipath elements and matrix switch cores. The operation of such elements is described.

273
Millimicrosecond Magnetic Switching and Storage Element, by D. A. Meier (The Natl. Cash Register Co.); *J. Appl. Phys.*, suppl. to vol. 30, pp. 45S–46S; April, 1959.

A magnetic rod suitable for performing the logical switching and storage functions required in a digital computer is discussed. The magnetic element consists of a silver-coated glass rod upon which is electroplated a Fe-Ni alloy several thousand angstroms thick. A single element for a coincident-current memory requiring two inputs, an inhibit winding, and a sense winding would consist of four separate single-layer concentric solenoids wound over the magnetic rod. Memory matrices, each consisting of many solenoids, can be stacked and simultaneously threaded with the rod. The switching speed of the element operating in a coincident-current mode is approximately 70 mμsec. The output voltage generated across a ten-turn sense winding is 200 to 500 mv depending upon the alloy thickness. The rod is also suitable as a multi-input logical switch. Separate inhibiting windings wound over the rod perform the NOR function of its input literals. A total of thirty separate inhibiting windings have been demonstrated in the laboratory. Satisfactory operation in the 2- to 5-mc range using transistors in conjunction with the rod has been shown to be practical. Continuous plating and testing in conjunction with automatic machine winding techniques make the rod appear economically attractive in digital computer applications. [See Abstract 134.]

274
High-Speed Switching Diodes from Plastically Deformed Germanium, by G. L. Pearson and R. P. Riesz (Bell Tel. Labs., Inc.); *J. Appl. Phys.*, vol. 40, pp. 311–312; March, 1959.

A study of the high-speed switching response of diffused diodes fabricated from plastically deformed germanium is reported. The degradation of minority-carrier lifetime caused by dislocations generated during plastic deformation greatly reduced the minority-carrier storage effect and permitted fabrication of diodes with turnoff times of the order of 10^{-9} sec.

275
A New Memory Device—The Twister, by D. A. Ellerbruch (Boeing Airplane Co.); IRE Trans. on Components Part, vol. CP-6, pp. 42–44; March, 1959.

Three versions of the "Twister" memory are discussed. The construction and operation of each are described.

276
Superconductive Devices, by A. E. Slade and H. McMahon (Arthur D. Little, Inc.); *Proc. WJCC*, pp. 103–107; May 6–8, 1958.

Following a discussion of superconductivity, the cryotron, a high-speed electronic switch which utilizes the nonlinear dependence of resistance on magnetic field in superconductors, and memory devices which utilize the ability of superconductors to carry a persistent current are discussed. The present limitations of wire-wound cryotrons are pointed out and cryotrons fabricated by vacuum deposition techniques which overcome these limitations are described. Persistent-current memory devices which have switching times as short as 10 mμsec have been reported.

277
A Cryogenic Oscillator, by G. B. Rosenberger (IBM Corp.); *IBM J. Res. & Dev.*, vol. 3, pp. 189–190; April, 1959.

A superconducting oscillator which is much simpler in design that those based on tank circuits is described. The oscillator depends on a relaxation process in the transition between the superconducting and conducting phases of a lead film which is shunted with a copper wire.

278
A New High-Speed Digital Technique for Computer Use, by D. Eldridge (Ferranti Ltd.); *Proc. IEE (London)*, vol. 106, pt. B, pp. 229–236; March, 1959. (Discussion pp. 237–239.)

A new method for realizing logical functions using square-loop ferrite cores and transistors is described. Complex circuits can be built up from identical elements, each consisting of one core, one transistor, three diodes, and one resistor. The technique differs from earlier methods in that the current required to set a core to the "1" state is derived from a common supply and not from a previous stage. The control of digit transfer between stages is effected by a transistor whose collector and emitter are connected across one winding of the core. When a "1" is to be set in the core, the transistor is arranged to appear as an open-circuit, in which condition flux reversal is possible; when a "0" is to be set, the transistor appears as a short-circuit preventing reversal of the flux. The core is reset to "0" at a controlled rate, producing a standardized voltage across the output windings. Logical operations are carried out by the analog addition of these voltages. The system requires only two low-voltage dc supplies and can stand voltage and component variations. It operates reliably at digit rates of 500 kc. An experimental application of the method is described.

279
Symmetrical Back-Clamped Transistor Switching Circuits, by R. H. Baker, R. E. McMahon, *et al.* (M.I.T. Lincoln Lab.); *U. S. Gov. Res. Repts.*, vol. 31, p. 238(A), April 17, 1959; PB 130 883.

A group of circuits developed for use in digital data-processing equipment is described. The circuit configurations are symmetrical and employ back-clamping where high rapid-circuit response is important. The use of these techniques yields circuits that are fast and very stable, have high efficiency, and require minimum specification of transistor and circuit parameters. The circuits yield maximum utility from the standpoint of building digital systems.

280
Doubling the Efficiency of the Load-Sharing Matrix Switch, by M. P. Marcus (IBM Corp.); *IBM J. Res. & Dev.*, vol. 3, pp. 194–196; April, 1959.

A method for doubling the efficiency of a load-sharing matrix switch is described. The efficiency is doubled by considering the inputs separately instead of in pairs. The number of drivers and associated equipment is halved, and with 2^n inputs, 2^n-1 instead of 2^{n-1} outputs are obtained.

281
A Generalized Resistor-Transistor Logic Circuit and Some Applications, by S. C. Chao (Link Aviation, Inc.); IRE Trans. on Electronic Computers, vol. EC-8, pp. 8–12; March, 1959.

A generalized resistor-transistor logic circuit, *i.e.*, a circuit for which the output produces a signal when any m out of the n inputs are "on," is discussed. Practical limitations such as the use of precision power supplies and components are considered. However, for small values of n and m, circuits could be designed such that no special precision components and supplies would be required. Several practical circuits are worked out, including a two-transistor binary full adder, a three-transistor comparator, and a one-transistor-per-bit-ring counter. These circuits, especially the first two, are uniquely simple and low in cost. They can be incorporated with other circuits to simplify a digital system. It is felt that with ordinary supplies (less than 5 per cent voltage variation) and 1 to 5 per cent resistors, these circuits can be designed to be very reliable. Experimental circuits employing germanium-alloy junction transistors have operated successfully at pulse rates up to 500 kc and an ambient temperature of 55°C.

282
Design of High-Performance Instrument Servos for General Purpose Computation, by K. V. Bailey and M. A. Ziniuk (Bendix Aviation Corp.); IRE Trans. on Industrial Electronics, no. PGIE-9, pp. 1–13; April, 1959.

A computing servo capable of accurate rate and position operation with extended frequency response relative to conventional computing servos is described. The usual

tachometer generator with its penalizing inertia is avoided. Rate feedback is provided instead, by a precision capacitor which couples the follow-up potentiometer to the input summing junction. Design objectives, compromises, and performance data are reviewed.

A-3: EQUIPMENT—SUBSYSTEMS

283
Binary Arithmetic for Discretely Variable Word Length in a Serial Computer, by P. Ercoli and R. Vacca (C.N.R., Rome); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 13–16; April, 1959.

A procedure whereby the arithmetic unit of a binary serial computer may conveniently be altered to provide for double-precision arithmetic is described. Adjusting the hardware is shown to be faster and more economical than programming for double precision. The machine employed is a Ferranti Mark I, but the technique is of general application.

284
Reversible, Diodeless, Twistor Shift Register, by A. H. Bobeck and R. F. Fischer (Bell Tel. Labs., Inc.); *J. Appl. Phys.*, suppl. to vol. 30, pp. 43S–44S; April, 1959.

A twistor shift register, which utilizes the interaction effects existing between magnetized regions on a magnetic wire, and which requires only a single magnetic wire and no diodes, is described. The information is stored as magnetically polarized zones which can be moved along the wire by means of a five-phase pulse source. Since no diodes are required and the only threshold consideration is the magnetic material itself, drive powers can be greatly decreased. Bi-directional operation is easily secured. The upper frequency limit has not been established; however, a several-hundred-kilocycle bit rate should be possible. Physically, the register could be made of no more than magnetic and copper wire. This should make fabrication considerably cheaper than for conventional shift registers.

285
Quadratic Interpolation in Tapped-Potentiometer Function Generators, by E. M. Deeley (Univ. of London); *Proc. IEE (London)*, vol. 106, pt. C, pp. 102–107; March, 1959.

A method for obtaining quadratic interpolation in a tapped-potentiometer generator for functions of a single variable is described. The method depends on the quadratic form of the variation of resistance to ground at the slider, across which the voltage necessary to reproduce correctly the first three terms of the Taylor series is developed. The supply to the tapping points from voltage sources of zero and finite resistance is analyzed and the problem of supplying these voltages is examined. The application of the method to the generation of functions of two variables is briefly discussed.

286
System Design of the Flying Spot Store, by C. W. Hoover, Jr., G. Haugk, and D. R. Herriott (Bell Tel. Labs., Inc.); *Bell Sys. Tech. J.*, vol. 38, pp. 365–401; March, 1959.

The factors which control the speed, capacity, number of channels, physical size, and probability of error in readout of a flying-spot store are discussed. The flying-spot store is a versatile and efficient, semipermanent information storage system developed for use in an electronic telephone switching system. It is shown that these factors are related to the intended application for the store, to the form of the store, and to device parameters. A number of scaling operations which involve exchanges between system parameters such as speed and capacity or between system and device parameters are given.

287
Optics and Photography in the Flying Spot Store, by M. B. Purvis, G. V. Deverall, and D. R. Herriott (Bell Tel. Labs., Inc.); *Bell Sys. Tech. J.*, vol. 38, pp. 403–424; March, 1959.

Some of the optical and photographic problems to be considered in the construction of a flying-spot store, a semipermanent binary information storage system in which a cathode-ray tube display is imaged on photographic emulsion by parallel optical channels, are discussed.

288
Beam-Positioning Servo System for the Flying Spot Store, by L. E. Gallaher (Bell Tel. Labs. Inc.); *Bell Sys. Tech. J.*, vol. 38, pp. 425–444; March, 1959.

The cathode-ray tube beam-positioning servo system, essential to the accurate and reliable microsecond access to the photographic information in a flying-spot store, is described. The characteristics of both the basic servo loop and its components are discussed, and several variations of the basic scheme are compared in relation to the system requirements.

289
Stable High-Speed Digital-to-Analog Conversion for Storage Tube Deflection, by D. F. Ault (Bell Tel. Lab., Inc.); *Bell Sys. Tech. J.*, vol. 38, pp. 445–465; March, 1959.

The design of access circuitry for a barrier grid-tube temporary memory is discussed. The circuitry converts a 14-bit binary address into the analog deflection voltage necessary to deflect the electron beam in the barrier grid tube to a specific geometrical storage area defined by the address. A special feedback circuit and raster reference tube, deflected in parallel with the barrier grid tube, control the size and centering of the array of storage spots. Novel methods of measurement were developed to certify the accuracy of the deflection system. The system meets the requirements of high speed, accuracy, stability, and reliability.

290
Coincident-Current Nondestructive Readout from Thin Magnetic Films, by L. J. Oakland (Remington Rand Univac); *J. Appl. Phys.*, suppl. to vol. 30, pp. 54S–55S; April, 1959.

Memory arrays built by using cores which consist of two-vacuum-deposited ferromagnetic films and from which data can be read out nondestructively by use of coincident-current selection are discussed. It is thus possible to eliminate the external selection matrices commonly associated with nondestructive readout memories. This system promises to be inexpensive, reliable, and fast, especially in a memory in which read operations occur more frequently than write operations. Three different devices which fulfill the criteria of coincident-current, nondestructive readout are described. Good signal-to-noise has been achieved with each of these.

291
Operating Characteristics of a Thin Film Memory, by J. I. Raffel (M.I.T. Lincoln Lab.); *J. Appl. Phys.*, suppl. to vol. 30, pp. 60S–61S; April, 1959.

An experimental prototype memory with 32 ten-bit words is discussed. Circular spots 1/16 inch in diameter, about 600-A thick, are used. These are evaporated on two pieces of glass each comprising a 16×16 spot array. An operating-cycle time of less than one-half microsecond appears possible. The circuitry for driving and sensing is transistorized and the memory uses external register selection from a core-diode matrix. Word selection is provided by a transverse field, and a digit winding conditions the information written by applying a longitudinal field in the "one" or "zero" direction. Extension to sizes of the order of 1000 words is planned using these techniques. The memory will soon be installed in the control element of the TX-2 computer.

292
Reversible Component of Magnetization, by R. W. McKay (Univ. of Toronto); *J. Appl. Phys.*, suppl. to vol. 30, pp. 56S–57S; April, 1959.

A method of nondestructive sensing of a ferrite-core memory which depends on the variation of the reversible component of magnetization with the state of the core has been suggested previously. Measurements of this reversible component were made for two types of cores, and it was found that the reversible component varies nearly three to one between the remanent state and the demagnetized state. The effects of finite core-wall thickness and of flux-leakage are discussed.

293
A Fast Method of Reading Magnetic-Core Memories, by H. J. Heijn and N. C. de-Troge (N. V. Philips' Gloeilampen Fabrieken); *Philips Tech. Rev.*, vol. 20, no. 7, pp. 193–207; 1958/1959.

Following a review of various types of memory systems, in which magnetic-core memories are discussed in detail, a memory system which utilizes cores of ferroxcube 6D3, a material having a rectangular hysteresis loop, and which contains 1024 words of 44 bits, is described. A feature of the memory is its low cycling time of about 3 μsec. This is achieved by preventing parasitic pulses from blocking the output amplifiers. Each output amplifier utilizes a transistor as a temporary storage element. A method used in the memory to produce current pulses of stable shape and height is described. The method is based on the saturation of a ferrite core in a choke.

294
Inhibited Flux—a New Mode of Operation of the Three-Hole Memory Core, by J. A. Baldwin, Jr., and J. L. Rogers (Bell Tel. Labs., Inc.); *J. Appl. Phys.*, suppl. to vol. 30, pp. 58S–59S; April, 1959.

A method of operation of the three-hole memory core is discussed. As is the case with the coincident-flux scheme developed by IBM, read and write times may be made very short. However, by using single-line drive rather than coincident current, one need pass but one conductor through each hole. A description is given of a small memory array which uses this method.

295
Transistor Core Memory, by R. E. McMahon and F. L. McNamara (M.I.T. Lincoln Lab.); *U. S. Gov. Res. Repts.*, vol. 31, p. 240(A), April 17, 1959; PB 130 881.

A transistor core memory which exhibits an appreciable reduction in space and power over tube-driven memories is described. The memory, which has a storage capacity of 4096 32-bit words, occupies 11 cubic feet and requires 550 watts of power. Circuit details and the "delta noise" problem are discussed. A complete evaluation of the memory relative to transistor failures and operating errors is included.

296
A High-Speed Analog to Digital Converter, by D. Savitt (Bendix Aviation Corp.); IRE Trans. on Electronic Computers, vol. EC-8, pp. 31–35; March, 1959.

An electronic voltage encoder which converts analog voltages to their corresponding parallel seven binary-digit representations at a 50-kc encoding rate is described. The encoder is capable of being time-shared by any number of 0–50-volt range inputs. Performance tests indicate that the present design may be capable of eight binary-digit conversions at encoding rates as high as 80 kc. Either more precise conversions or higher encoding rates may be obtained at the expense of the other by cascading more or less of the identical one-digit encoder stages which constitute the analog to digital converter.

297
A Figure of Merit for Single-Pass Data Recording Systems, by J. H. Mulligan, Jr. (New York Univ.); IRE Trans. on Electronic Computers, vol. EC-8, pp. 48–54; March, 1959.

The problem of the interference caused by eddy current transients to the reproduction of recorded data is studied for single-pass magnetic recording systems of both the write-read and read-write variety. Signal-to-interference ratios are introduced for both modes of operation, and their variation is studied in detail. It is found convenient to introduce a dimensionless parameter a as a figure of merit for single-pass systems. This factor is a function of the velocity of the magnetic medium, the number of pulses that can be recorded per unit length, and the permeability, conductivity, and dimensions of the laminations of the recording head. Brief consideration is given to the effect of certain practical system factors on the conclusions reached from the theoretical analysis.

298
A High-Speed Electronic Printer for Digital Computer Output, by W. S. Grant (Radiation, Inc.); *U. S. Gov. Res. Repts.*, vol. 31, p. 161(A), March 13, 1959; PB 151 331.

The design and development of a high-speed electronic printer for a digital computer output are discussed. The electrical and mechanical design requirements of the equipment are pointed out and the resulting circuit design is described. The system consists of a single timer unit, a single numeric-function generator, and twelve character elements. The character elements are identical and consist of the necessary circuitry to produce any decimal digit 0–9, a minus sign, and a decimal point. The printer writes twelve characters in parallel across a five-inch chart at a speed of 2160 characters per second and uses a 6-bit binary code input based on the Flexowriter system from an ERA 1103 Computer. Operating procedures and future design considerations are included.

299
A Sensing System for Punched Cards or Continuous Punched Foil, by S. Morleigh; *Electronic Engrg.*, vol. 31, pp. 140–141; March, 1959.

A capacitive sensing method for determining the location of any hole in a punched card is described. The punched card consists of a dielectric medium, one side of which has a continuous metallic coating, and the other side of which has conducting strips whose positions correspond to the column positions of a standard card. The two capacitances formed when the conducting strip is broken by a punched hole (the width of the hole is larger than that of the strip) are used as two arms of a capacitance bridge. The output of the bridge indicates the position of the hole. The advantages of the method are pointed out.

300
Random Number Generator Using Subharmonic Oscillators, by F. Sterzer (RCA); *Rev. Sci. Instr.*, vol. 30, pp. 241–243; April, 1959.

A random number generator which uses two 2000-mc subharmonic oscillators is described. The generator can produce random binary digits at a maximum rate of approximately 3×10^7 digits per second. Statistical tests performed on over a quarter-million digits produced by this generator indicate that they are at least as random as digits produced by other slower contemporary means.

A-4: EQUIPMENT—DIGITAL COMPUTERS

301
The X-1 Computer, by B. J. Loopstra (N. V. Electrologica); *Computer J.*, vol. 2, pp. 39–43; April, 1959.

The X-1, a small, fast, transistorized computer, developed at the Mathematical Centre, Amsterdam, is described. A fixed-point binary, single-address machine of 27-digit word length and core memory, it is intended primarily for commercial use.

302
The New Univac Computer; *Univac Rev.*, vol. 2, pp. 3–7; Winter, 1959.

The new, completely solid-state Univac Computer is described in detail. Utilization of the latest developments in magnetic cores, Ferractor amplifiers and transistors produces a compact high-speed machine of high reliability, with significantly low power consumption. Careful tailoring of peripheral equipment results in a highly integrated data-processing system. Primarily intended for business application, the machine can also handle scientific and engineering calculations conveniently. The new Flow-Matic coding simplifies programming.

303
The National Physical Laboratory's Ace; *Computer Bull.*, vol. 2, pp. 79–80; February–March, 1959.

A brief description of the British National Physical Laboratory's new Automatic Computing Engine or ACE is given.

304
Operating Experience with a Transistor Digital Computer, by R. C. M. Barnes and J. N. Stephen (U.K.A.E.A. Atomic Energy Res. Establ.); *Proc. IEE (London)*, vol. 106, pt. B, pp. 222–228; March, 1959.

The performance over a period of a year of Cadet, a small digital computer which uses transistors as the only active circuit elements, is described. Failure rates of 1.2 per cent per 1000 hours were observed for point-contact transistors and 0.5 per cent for a small sample of junction transistors. Earlier information on this computer is supplemented by details of circuit developments.

305
Machine Recognition of Hand-Sent Morse Code, by B. Gold (M.I.T. Lincoln Lab.); IRE Trans. on Information Theory, vol. IT-5, pp. 17–24; March, 1950.

The design and operation of a transistorized special purpose digital computer called MAUDE (Morse AUtomatic DEcoder) which decodes a hand-sent Morse code message and prints it on a teletypewriter are discussed. The performance of MAUDE is analyzed.

A-5: EQUIPMENT—ANALOG COMPUTERS

306
Electronic Differential Analyzers in Perspective, by J. McLeod (Convair Astronautics); *Proc. WJCC*, pp. 82–86; May 6–8, 1958.

Following a brief description of the solution of a problem by means of an analog computer, the advantages and disadvantages of analog computers are discussed.

307
Simultaneous Gain-Phase Approximation with a Potential Analog Computer, by P. M. Liebman (Polytech. Inst. of Brooklyn); *U. S. Gov. Res. Repts.*, vol. 31, p. 145(A), March 13, 1959; PB 135 097.

The theory and operation of a potential analog computer capable of solving network synthesis problems are discussed. The computer may be used for simultaneous amplitude and phase synthesis and the design of All-Pass networks. The device is based on the electro-potential analogy and reduces the problem to one of locating line sources in a conducting medium. The line sources, depending on their polarity, represent the poles and zeros of the function to be approximated.

308
Analog Computer for Estimating Fire Endurance; *NBS Tech. News Bull.*, vol. 43, pp. 32–34; February, 1959. (See also *J. Res. NBS*, vol. 61, pp. 105–115; August, 1958.)

A special-purpose analog computer which utilizes a direct analogy between thermal and electrical networks to solve transient heat-flow problems in building materials is described. The instrument is of the so-called "fast-time" type and simulates the transient involved in times of the order of 10^{-6} that of the thermal prototype. A photoformer type of input signal generation is used and this, together with a variable-frequency master oscillator, permits flexibility of input signal waveform and time scale. The instrument has applications in the field of fire research.

B-1: SYSTEMS—THEORETICAL DESIGN

309
Organizing a Network of Computers; *Datamation*, vol. 5, p. 39, March–April, 1959; (See also *NBS Tech. News Bull.*, vol. 43, pp. 26–28; February, 1959.)

An investigation of the logical problems which arise when several high-speed electronic computers are connected together to work on a common large-scale task is discussed. It has been found that queuing-theory techniques can be applied to evaluate time lost when one machine in the network must stop and wait for another to finish its part of the task and to devise schemes for making computers in the network share a common workload. A simple model network, in which a primary computer shares its workload with a secondary computer connected to it via transmission lines, is described.

B-3: SYSTEMS—APPLICATIONS

310
Simulation to Obtain a Systems Measure of an Air Duel Environment, by A. A. B. Pritsker, R. C. Van Buskirk, and J. K. Wetherbee (Battelle Memorial Inst.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 55–59; March, 1959.

A combined analog-digital simulation of an air battle between an attacking bomber aircraft and a ground controlled interceptor, including the intermediate human radar operator, designed for the purpose of evaluating the effects of airborne electronic countermeasures upon a ground-based radar operator, is described. Both real and nonreal time simulation are used in the experimental setup. The simulation encompasses as much as possible of the system as affected by the operators' performance, in the hope that a systems measure could be obtained. It is hypothesized that probable success of the bomber is the systems measure of the effectiveness of the countermeasures.

B-4: SYSTEMS—TESTING

311
A Nonreal-Time Simulation of SAGE Tracking and BOMARC Guidance, by D. W. Ladd and E. W. Wolf (Mitre Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 36–41; March, 1959.

The addition of facilities to the SAGE system for control of a new defensive weapon, such as the BOMARC missile, requires extensive modifications to the SAGE computer program. To obtain a better understanding of BOMARC control problems, a program has been written for the IBM Type 704 computer to simulate the proposed employment of BOMARC in the SAGE system. Such a simulation is flexible enough to optimize and evaluate a large range of parameters. On three separate passes through the 704 (with tape storage of intermediate results) the program simulates radar, target, and missile performance, as well as SAGE tracking and missile guidance. A fourth program presents the desired output data in the form of frequency distributions and detailed results pertaining to selected target or missile tracks.

C-1: AUTOMATA—NATURAL

312
Characteristics of Simple Manual Control Systems, by J. I. Elkind (M.I.T. Lincoln Lab.); *U. S. Gov. Res. Repts.*, vol. 31, p. 240(A), April 17, 1959; PB 130 844.

A method for measuring and describing the characteristics of manual control systems is presented. The method is applied in an experimental study of the characteristics of simple manual systems. The experimental results are discussed and analytic models which approximate the measured characteristics are derived. An analog computer developed to implement the measurement of the system characteristics is described.

C-2: AUTOMATA—ARTIFICIAL

313
The Reduction of Two-Way Automata to One-Way Automata, by J. C. Shepherdson (Univ. of Bristol, Eng.); *IBM J. Res. & Dev.*, vol. 3, pp. 198–200; April, 1959.

A shorter, more direct proof is presented of Rabin's result that finite two-way automata are equivalent to one-way automata, as far as the classification of input tapes is concerned. The essence of the proof is that a machine need not refer back to a piece of tape, t, if before leaving t it answers all the questions it might later come back and ask about t, and carries the question-answer combinations along the tape as it goes along.

314
The Mechanization of Thought Processes; *Computer Bull.*, vol. 2, pp. 92–93; April–May, 1959.

The proceedings of a recent British Computer Society Symposium on mechanization of thought processes are reviewed. Information selection, medical diagnostics, character and speech recognition, and artificial intelligence are among the topics discussed.

315
Could a Machine Make Probability Judgments? by I. J. Good; *Computers and Automation*, vol. 8, pp. 14–16, January, 1959; pp. 24–26, February, 1959.

Using as an analogy the way human chess players make probability judgments in assessing their strategy, it is suggested that a computing machine with random networks may, within the foreseeable future, be making probability judgments as effective as those of experienced humans.

316
Finite Automata and Their Decision Problems, by M. O. Rabin (Hebrew Univ., Jerusalem) and D. Scott (Univ. of Chicago); *IBM J. Res. & Dev.*, vol. 3, pp. 114–125; April, 1959. (See also Review of this paper by T. H. Crowley, this issue, p. 407.)

Finite automata may be considered as a subclass of Turing machines having a finite number of internal states and finite tapes. In this they are similar in concept to nerve-sets. The abstract theory of finite automata, divided into one-tape, one-way; one-tape, two-way; and multi-tape automata, is discussed. Two-way automata are shown to be no more powerful than one-way; and the decision problem for a two-tape automaton is not solvable by an effective algorithm.

317
The Optimization of Logical Goal-Seeking Procedures, by A. Porter (Univ. of Saskatchewan) and P. K. T. Vaswani (Imperial College of Sci. and Tech.); *J. Electronics and Control*, vol. 6, pp. 168–185; February, 1959.

The property of decision-taking is basic in all types of control systems, and the goal-seeking characteristic of such systems can be considered as a direct consequence of this decision-taking capability. The difficulty of optimizing goal-seeking procedures in systems in which the rules, or constraints, are based on two-valued logical statements is discussed. It is shown that, if the variables in a particular system and the corresponding rules of behavior can be expressed in terms of a many-valued logic, a function, called the measure function, can be defined which provides a crude method of optimizing the goal-seeking process. The approach is essentially empirical. A high-speed computer is used to study selected 16-valued logical problems. The main object of the paper is to stimulate interest among control specialists in the possibility of applying many-valued logics in the control of complex processes.

D-1: PROGRAMS—AUTOMATIC PROGRAMMING, DIGITAL COMPUTERS

318
From Formulas to Computer Orientated Language, by J. H. Wegstein (Natl. Bur. of Standards); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 6–8; March, 1959.

A recursive technique for enabling a computer to translate simple algebraic formulae into a three-address code is described. The description is achieved by means of a flow diagram.

319
Zurich Conference on Algorithmic Language; *Computer Bull.*, vol. 2, pp. 81–82; February–March, 1959.

A reprint of the introductory part of the ACM-GAMM Zurich report on an International Algebraic Language is presented. The objectives of the IAL are: 1) it should be as close as possible to standard notation and readable with little explanation; 2) it should be adaptable for use in publications; and 3) it should be mechanically translatable into machine programs. IAL is developed on the three levels of Reference Language, Publication Language and Hardware Representations. (See abstract 179.)

320
Possible Modifications to the International Algebraic Language, by J. Green (IBM Corp.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 6–8; February, 1959.

A list of possible modifications to the IAL or "Zurich Language" is proposed. The modifications are considered to enhance it without adding unnecessary bulk. A comparison table of simple problems described both in IAL and with modifications is provided.

321
Signal Corps Research and Development on Automatic Programming of Digital Computers, by W. F. Luebbert and P. W. Collorm, Jr. (U. S. Army); *Commun. Assoc. for Computing Math.*, vol. 2, pp. 22–27; February, 1959.

The basic ideas of automatic programming and the simplification of coding are reviewed, and the Signal Corps work on these problems is described. Interpretative translation programs, which perform each part of program immediately on translation into machine language, are frequently encountered in the field of combat intelligence. The main effort is being directed towards providing a fast and efficient means of producing specialized compilers to handle limited technical vocabularies with a strong common language basis.

322
Use of Compiler Programs to Solve Power System Problems on a Large Digital Computer, by J. T. Carleton and C. J. Baldwin, Jr. (Westinghouse Elec. Corp.); *Power Apparatus and Systems*, no. 40 (*Trans. AIEE*, pt. III, vol. 78), pp. 1319–1324; February, 1959.

The use of compiler programs as the bridge between the language of a power system engineer and that of a computer is discussed. Compiler programming is explained and examples of its application are given. Performance data on its use are presented. The compiler program discussed is the FORTRAN System for the IBM 704 computer.

323
Recursive Subscripting Compilers and List-Type Memories, by J. W. Carr (Univ. of Michigan); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 4–6; February, 1959.

Algorithms for arranging a list in a particular sequence, for adding or deleting elements from a list and for finding the first occurrence of a certain value in a list of numbers are described. The technique used is that of recursive indexing. The operations described are of particular value for symbol manipulation, artificial intelligence, language translation, theorem proving, and the like.

324
The Air Force Breaks Through New Communications Barrier, by E. R. Miller (U.S.A.F.) and J. L. Jones (Dayton A.F. Depot); *Univac Rev.*, vol. 2, pp. 8–12; Winter, 1959.

AIMACO, an automatic programming technique developed by Air Materiel Command and Remington Rand, is described. Both Univac Flow-Matic Coding and Univac 1105 USE Compiler are utilized. The resulting pseudo-code accepts English words and phrases, which are then translatable into either Univac 1 or 1105 machine codes. Communication between management and programmers is improved and much programming drudgery avoided.

325
The SHARE 709 System: A Cooperative Effort, by D. L. Shell (G.E. Co.); *J. Assoc. for Computing Mach.*, vol. 6, pp. 123–127; April, 1959.

The pedigree of the 709 System, from the "Comprehensive System" of the M.I.T. Whirlwind through the 704 SHARE Assembly Program, is traced; and the cooperative effort that went into its making is evaluated.

326
The SHARE 709 System: Machine Implementation of Symbolic Programming, by E. M. Boehm (IBM Corp.) and T. B. Steel, Jr. (Syst. Dev. Corp.); *J. Assoc. for Computing Mach.*, vol. 6, pp. 134–140; April, 1959.

Between the high redundancy of a symbolic programming language and the machine language, a second "SQUOZE" encoding is introduced. The "SQUOZE" encoding contains all the information of the symbolic programming, but with far less redundancy, and hence the machine time spent in processing it is reduced.

327
The SHARE 709 System: Programmed Input-Output Buffering, by O. Mock (No. Amer. Aviation Corp.) and C. J. Swift (Convair); *J. Assoc. for Computing Mach.*, vol. 6, pp. 145–151, April, 1959.

The 709 Input-Output buffering necessary to enable input-output of information to take place simultaneously with computing is described.

328
The SHARE 709 System: Input-Output Translation, by V. J. DiGri (IBM Corp.) and J. E. King (G.E. Co.); *J. Assoc. for Computing Mach.*, vol. 6, pp. 141–144; April, 1959.

Various criteria for Input-Output Translation, such as memory space vs execution time, are evaluated, and the main MACRO instructions of the SHARE 709 translation system are described.

329
The SHARE 709 System: Supervisory Con-Control, by H. Bratman (Lockheed Aircraft Corp.) and I. V. Boldt, Jr. (Douglas Aircraft Co.); *J. Assoc. for Computing Mach.*, vol. 6, pp. 152–155; April, 1959.

A supervisory control program, responsible for keeping the computer continuously operating while performing a group of independent tasks, is described. In addition, the program takes care of accounting and other records.

330
The SHARE 709 System: Programming and Modification, by I. D. Greenwald (RAND Corp.) and M. Kane (IBM Corp.); *J. Assoc. For Computing Mach.*, vol. 6, pp. 128–133; April, 1959.

A general description of the operations of the SHARE 709 or SCAT System in handling the programming problems of changes and corrections, debugging, and translation from programmers' to machine language is given.

331
SESAME Opens the Door to Programming Simplification, by L. R. Johnson and R. D. Pratt (Remington Rand); *Univac Rev.*, vol. 2, pp. 13 and 16–17; Winter, 1959.

SESAME (SErvice Sort And MErge), a project to produce efficient and general sorting-merging programs for Univac II, is described. The project is designed to provide the user with flexible tools that take advantage of the favorable circumstances that characterize a given filing application, without sacrificing optimum speed on random input.

332
The Arithmetic Translator-Compiler of the IBM FORTRAN Automatic Coding System, by P. B. Sheridan (IBM Corp.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 9–21; February, 1959.

The steps employed by the FORTRAN arithmetic translator in translating into 704 assembly code are described in formal terms. The reader is assumed to be familiar with FORTRAN II as well as with SAP II and the logic of the 704 computer.

333
A Checklist of Intelligence for Programming Systems, by R. W. Bremer (IBM Corp.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 8–13; March, 1959.

An attempt is made to begin a systematic classification of the various devices, such as processors, supervisors, and languages, that help a computer to take over the decision-making functions of human operations. A Yes-No checklist of appropriate questions is followed by amplifying notes.

334
Blueprint for a Library, by B. H. Worsley (Univ. of Toronto); *Computers and Automation*, vol. 8, pp. 17–20; April, 1959.

The design of a program library for a medium-to-large computer in a university setting is discussed. Such a library should be on two levels; a flexible and primary system in the natural machine language, and a secondary scheme consisting of packages of routines. The number of student and research workers directly able to use the machine could then be considerably increased.

D-2: PROGRAMS—APPLICATIONS, DIGITAL COMPUTERS

335
Business Judgment Simulated through Management Simulation, by D. H. McGalliard (Remington Rand); *Univac Rev.*, vol. 2, pp. 20–21; Winter, 1959.

Problems of designing realistic mathematical models for business management simulations (business games) are discussed. For example, over-simplified models are insensitive to significant changes in policy; a too-complex model makes the isolation of cause-and-effect relationships impossible. Reasonable flexibility is also required. Models selected should be evaluated by the same criteria as used for scientific hypotheses.

336
University Student Scheduling by Datatron, by N. Sabbagh (Purdue Univ.); *Computers and Automation*, vol. 8, p. 13; March, 1959.

Purdue University's system of registering students with the aid of a computer, and the consequent savings in time and confusion, are discussed. A student's courses now are scheduled in a maximum of two-minutes computer time.

337
Preliminary Systems Design and Operational Analysis for the MIFD, by D. B. J. Bridges, J. E. Davis, *et al.* (Battelle Memorial Inst.); *U. S. Gov. Res. Repts.*, vol. 31, p. 240(A), April 17, 1959; PB 136 234.

The first constructive work done in developing a Materiel Information-Flow System for use with the Materiel Information-Flow Device in an experimental installation is reported. It is proposed that the computer be programmed to scan the remotely connected input devices, to process requests for material after checking against stock records held in the computer memory, and to print the proper issue orders on output units located in a warehouse. The initial programming for the system is included.

338
Calculation of Transmission-Line Impedances by Digital Computer, by A. O. Thomas (Pennsylvania Power and Light Co.); *Power Apparatus and Systems*, no. 40 (*Trans. AIEE*, pt. III, vol. 78), pp. 1270–1274; February, 1959. (Discussion pp. 1274–1275.)

The calculation of positive- and zero-sequence and mutual impedances of transmission lines on the IBM 650 digital computer is discussed. The discussion includes research to obtain basic physical transmission-line data, the method used to combine and list large quantities of data for card punching, the major components of the program, and the final tabulation of impedances.

339
Digital Calculation of Overhead-Transmission-Line Constants, by D. Coleman, F. Watts, and R. B. Shipley (TVA); *Power Apparatus and Systems* no. 40 (*Trans. AIEE*, pt. III, vol. 78), pp. 1266–1268; February, 1959. (Discussion pp. 1268–1270.)

A method for the calculation of overhead transmission line constants is described, and a program for the method which has been written for an IBM 704 computer is discussed. The method permits the calculation of individual conductor impedances and capacitances, both self and mutual, the positive-, negative-, and zero-sequence self- and mutual-inductances and capacitances of 3-phase, 60-cycle power circuits of any configuration, regardless of bundling and ground-wire arrangements for single or parallel lines. The method utilizes tensor analysis methods and is discussed in enough detail so that it can be programmed for machines other than the IBM 704. The program requires a minimum of data preparation.

340
The IBM 650 as a Tool for Analysis of Transmission and Distribution System Problems, by G. E. Adams and J. E. Gerngross (G.E. Co.); *Power Apparatus and Systems*, no. 40 (*Trans. AIEE*, pt. III, vol. 78), pp. 1236–1244; February, 1959.

The use of the IBM 650 Computer to solve a number of specific problems dealing with the performance of transmission and distribution systems is discussed. In each case, the procedure is described in sufficient detail to permit a program to be set up for a similar problem. Computer techniques are emphasized, and the details of the calculations and computer programs are given.

341
Nodal Representation of Large Complex-Element Networks Including Mutual Reactances, by J. C. Siegel and G. W. Bills (No. Amer. Aviation, Inc.); *Power Apparatus and Systems*, no. 40 (*Trans. AIEE*, pt. III, vol. 78), pp. 1226–1228; February, 1959. (Discussion pp. 1228–1229.)

A method for the calculation of steady-state currents and voltages in electrical power systems which have complex elements including mutual reactances is described. Nodal representation is employed, and the solution is obtained with the aid of digital computers by iteration. A sample problem and a flow chart for a program prepared for an IBM 704 digital computer are presented.

342
Forecasting Procedures Advance Effective Water Routings on the U. S. Columbia River Hydroelectric System, by H. N. McIntyre and M. S. Sachs (Bonneville Power Admin.); *Power Apparatus and Systems*, no. 40 (*Trans. AIEE*, pt. III, vol. 78), pp. 1588–1593; February, 1959. (Discussion pp. 1593–1595.)

The development and application of dependable stream-flow forecasting procedures for more effective scheduling and routing of the water resources available to the U. S. Columbia River Power System are described. The established procedures, which are facilitated by the use of a general-purpose digital computer, constitute an important step toward obtaining maximum utilization of the water resources for power production. The computer speeds the forecast calculations and permits quicker development of water routings for current system operations. Management is thereby given assurance to permit the sale of additional secondary power with adequate protection that subsequent system firm-power requirements will be fully met.

343
Verification of the Logic Structure of an Experimental Switching System on a Digital Computer, by D. G. Leagus, C. Y. Lee, and G. H. Mealy (Bell Tel. Labs., Inc.); *Bell Sys. Tech. J.*, vol. 38, pp. 467–476; March, 1959.

An IBM 704 program which was constructed to verify the logical structure of an experimental electronic switching system is described. The program simulates the operations of the central control, the scanner, and the network control of the experimental switching system; carries out the switching system program orders stored in a flying spot store; and gives output indications of whether calls submitted to the system are successfully completed.

344
Some Helicopter Simulation Studies, by J. M. Harrison (Westland Aircraft, Ltd); *Computer J.*, vol. 2, pp. 10–23; April, 1959.

The simulation of a single-main-rotor helicopter on a digital computer is discussed. Main features, illustrated by flow-charts of the required routine, include the craft's response in longitudinal mode during take-off and power-off landing, and the reaction of the propulsion system to the pilot's demands.

345
Weather Prognosis by Univac, by J. W. Mauchly (Remington Rand), *Univac Rev.*, vol. 2, pp. 18–19; Winter, 1959. (Reprinted from *Almanack*, IRE Philadelphia Sec., May, 1958.)

Weather forecasting by means of computers is discussed. Current meteorological use of computers is restricted to 24- to 36-hour forecasting of barometric pressures. From this data the meteorologist must draw his own conclusions. By experimentation with more complex mathematical models of the atmosphere, more accurate techniques are being developed. Improved long-range forecasting is one of the goals anticipated from the introduction of computers.

346
Memory Efficiency, by G. S. Joachim (Minn. Dept. of Highways); *J. Assoc. for Computing Mach.*, vol. 6, pp. 172–175; April, 1959.

Ten typical routines are analyzed and the percentages of memory-inactive time are computed. In these cases, memory proved to be active for only 35 per cent of computer running time. It is pointed out that more efficient organization could free memory for other purposes, *e.g.* it could be used for input-output during periods of inactivity.

347
New Method for the Statistical Computation of Polymer Dimensions, by F. T. Wall and J. J. Erpenbeck (Univ. of Illinois); *J. Chem. Phys.*, vol. 30, pp. 634–637; March, 1959.

A new method for the generation of excluded-volume random walks of contour lengths comparable to those of real polymer molecules is described. The walks were generated on the Illiac digital computer. An essential feature of the method is an unbiased sample enrichment process used to counteract the attrition resulting from chain intersections. Using the new method, samples were generated for chains of 800 links, a limit imposed by machine storage capacity. In principle, the method could be carried even further if more storage were available.

348
Structure Factor Calculations for Some Helical Polypeptide Models, by D. R. Davies and A. Rich (Nat. Inst. of Mental Health); *Acta Crystallographica*, vol. 12, pp. 97–101; February, 1959.

The calculation of the structure factors of some helical polypeptide models, assuming random angular orientations of the molecules about the helical axis, is discussed. A brief description of the method of computation, which utilizes punched cards and the IBM 604 computer, is given.

349
The Study of the Application of a Computer to Production Control, by D. C. Hemy and W. J. Kease (E.M.I. Electronics Ltd.); *Computer J.*, vol. 2, pp. 24–38; April, 1959.

Existing production control systems are reviewed and the required characteristics of a computer intended for such applications are enumerated. A system of production and

material control which makes use of minimax criteria where appropriate is outlined. The problems of translating proposed production into manufacturing and material delivery schedules are discussed.

350
Pattern Recognition by Means of Automatic Analogue Apparatus, by W. K. Taylor (University College, London); *Proc. IEE (London)*, vol. 106, pt. B, pp. 198–209; March, 1959.

The problem of synthesizing apparatus that will automatically simulate man's ability to recognize and to learn to recognize patterns is discussed. It is concluded that analog, rather than digital, switching circuits that have been employed in the past, provide the simpler solution. A new circuit unit, possessing many of the essential functional characteristics exhibited by nerve cells in the brain, is derived from earlier work on the electrical simulation of nervous-system functional activity and forms the basic element of the circuits. The new analog apparatus consists of a number of distinct functional circuits arranged in a definite sequence, through which signals derived from the patterns to be recognized pass simultaneously on their way to the final output terminals. Classification information may be built into the apparatus initially if it is available, but otherwise it can be stored automatically in a special unit during a setting-up procedure in which samples of the pattern types that the apparatus will be required to recognize are presented, together with the identification signals. Low-resolution automatic pattern-recognition apparatus is described, and examples which illustrate the setting-up procedure and subsequent performance of the apparatus are presented.

351
A System for the Automatic Recognition of Patterns, by R. L. Grimsdale, F. H. Sumner, C. J. Tunis, and T. Kilburn (Univ. of Manchester) *Proc. IEE (London)*, vol. 106, pt. B, pp. 210–221; March, 1959.

A method for the automatic recognition of patterns such as hand-drawn capital letters and numerals is described. Although the method may be applied to any form of spatial pattern, only patterns consisting of line figures are considered. The pattern is presented to a flying-spot scanner connected to a digital computer. The shape of the pattern is analyzed and a statement describing the basic features of the pattern is prepared. The pattern is then recognized by comparing this statement with a number of others already stored in the computer. Patterns are recognized independently of the angle at which they are presented to the scanner and may be of any size provided that limits imposed by the resolution of the scanner are not exceeded. The average time to recognize a character is 60 seconds with the system programmed on a medium-speed computer. Special-purpose equipment built to perform certain stages of the process, together with the use of higher-speed computers, will reduce this time. Special allowances are made for imperfections in the patterns. If an unknown pattern resembles two or more standard patterns, the relative degrees of similarity of the unknown to each of these standard patterns is printed out by the machine. A new pattern can become a standard pattern which the machine will recognize if its name is given to the machine.

352
A Chess Playing Program for the IBM 704, by A. Bernstein and M. deV. Roberts (IBM Corp.) and T. Arbuckle and M. A. Belsky (Service Bureau Corp.); *Proc. WJCC*, pp. 157–159; May 6–8, 1958.

The program which permits a human being to play a game of chess with the IBM 704 is described. The machine may play either white or black, and is capable of playing a complete game. The program is divided into five parts: (1) Input-output, (2) Table generation, (3) Evaluation, (4) Decision, and (5) Tree.

353
The Design and Operation of the Mechanical Speech Recognizer at University College, London, by P. Denes (University College); *J. Brit. IRE*, vol. 19, pp. 219–229; April, 1959. (Discussion pp. 230–234.)

The design of a practical speech recognizer is described. The recognizer consists of an acoustic spectrum analyzer; a spectral pattern matcher; a store of information about linguistic statistics; a phoneme sequence memory; a computer; and output circuits for the storage of computer decisions and the control of an automatic typewriter. The speech material used with the completed recognizer is described together with the results obtained. Future developments in speech typewriter research and possible uses of a successful recognizer are indicated.

D-3: PROGRAMS—TECHNIQUES, DIGITAL COMPUTERS

354
Radix Exchange—an Internal Sorting Method for Digital Computers, by P. Hildebrandt and H. Isbitz (Syst. Dev. Corp.); *J. Assoc. for Computing Mach.*, vol. 6, pp. 156–163; April, 1959.

An exchange method for internal sorting of items represented in binary form is described. The method is considerably faster than inserting for random data, compares favorably with internal merging, and has the advantage of requiring no working memory space.

355
A Routine to Find the Solution of Simultaneous Linear Equations with Polynomial Coefficients, by E. H. Larson and D. P. Marshall (Convair); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 16–17; April, 1959.

A routine for solving simultaneous linear equations with polynomial coefficients, based on Jordan elimination in which all polynomial divisions are exact, is described. The accuracy of the routine is somewhat limited.

356
Techniques for the Recording of, and Reference to Data in a Computer, by A. S. Douglas (Univ. of Leeds); *Computer J.*, vol. 2, pp. 1–9; April, 1959.

The basic techniques for storing labelled data in a computer are summarized and their relative merits compared. Various methods of rearrangement of and reference to the data, such as explicit and implicit labelling, sorting, merging, indexing, packing, and marking are discussed in considerable detail, with particular reference to the number of comparisons required in each case.

D-4: PROGRAMS—TESTING, DIGITAL COMPUTERS

357
Test Programs for HEC, by A. D. Woolner (Internatl. Computers and Tabulators Ltd.); *Computer J.*, vol. 2, pp. 44–47; April, 1959.

Test programs for initial testing, routine maintenance, and error diagnosis on the Hollerith type 1201 (or HEC-4) Computer are described.

D-5: PROGRAMS—APPLICATIONS, ANALOG COMPUTERS

358
Stabilization of Computer Circuits, ed. by E. Hochfeld (Univ. of Chicago); *U. S. Gov. Res. Repts.*, vol. 31, p. 97(A), February 13, 1959; PB 151 255.

Continuous computers are frequently used to solve a set of differential equations or to generate known functions as solutions to a set of differential equations. In general, the output errors in a continuous computer are time-dependent and will tend to grow. This places a limitation on the running time of most analog problems. It is shown how the functions in question can be generated as solutions to a set of differential equations different from the classical set. The new set of differential equations includes corrective factors which are functions of the errors; and, though the circuit which represents their solution is more intricate than the classical circuit and involves more components, its solutions will have error bounds that are not time-dependent. This basic theorem is proved. As specific applications of the general theory, it is shown how one can generate the trigonometric functions sine and cosine and a set of three orthonormal vectors wherein the errors in the solutions will have bounds which are time-invariant. Experimental evidence for the above is included.

359
Use of An Electronic Analogue Computer with Resistance Network Analogues, by J. Ph. Korthals Altes (Delft Univ. of Tech.); *Brit. J. Appl. Phys.*, vol. 10, pp. 176–180; April, 1959.

A method of solving partial differential equations of the elliptical type with an electronic analog computer in conjunction with a resistance network is described. Although the method is iterative, the adjustments are made automatically by means of electronic memory elements and a switching mechanism. Some results are given. If the iteration process is divergent, an additional voltage source connected with a well-chosen mode may offer a solution.

360
Simulating Second-Order Equations, by D. G. Chadwick (Utah State Univ.); *Electronics*, vol. 32, p. 64; March 6, 1959.

A method which utilizes only one operational amplifier to simulate a second-order differential equation on an analog computer is described.

361
SAINT Computer, An Analysis, by J. A. Lyden, Jr. (A. F. Cambridge Res. Center); *U. S. Gov. Res. Repts.*, vol. 31, p. 217(A), April 17, 1959; PB 136 045.

The SAINT (Semiautomatic Analog INTercept computer), which can direct an attack at any heading relative to a target, is discussed. The computer predicts the entire intercept path and displays this prediction on a PPI. The prediction is an integral part of the computation. The simultaneous predictions of several computers can be used, visually and/or automatically, to determine possible collisions with other aircraft or terrain obstacles. The computer operation has been studied analytically, and a laboratory model has been tested under simulated conditions to verify and to complement the analytical study. The studies have been primarily concerned with determining the fundamental computer operations to be performed and the difficulties involved in performing them. The results of these studies indicate that the described method of intercept computation is feasible and may have considerable operational utility.

362
Grid-Current Measurements with a D-C Analog Computer, by J. R. Cox (A and M College of Texas); *Electrical Engrg.*, vol. 78, pp. 162–165; February, 1959.

The measurement of grid currents which are beyond the range of ordinary measuring devices by means of d-c analog-computer elements is discussed.

363
The Response of a Flat Simply-Supported Panel to the Action of a Pressure Wave, by G. Isakson and C. W. Brenner (Mass. Inst. Tech.); *U. S. Gov. Res. Repts.*, vol. 31, p. 101 (A), February 13, 1959; PB 135 319.

The response of a flat uniform panel, simply supported along two parallel edges and unsupported along the other two, to the passage over it of a pressure wave of typical form is analyzed. Two cases are considered—one in which the wave impinges on the panel perpendicularly, and the other in which the wave moves parallel to the panel. The problem is set up in terms of non-dimensional parameters, and a Lagrangian type of analysis is carried out. A program is prepared for solution by means of an electronic analog computer of the differential-analyzer type. The results of such computations carried out on the Reeves Electronic Analog Computer (REAC) are presented for practical ranges of of the significant parameters. A comparison is made of results for the two cases, and it is found that it is not feasible to introduce the simplification of replacing the parallel-moving wave by an equivalent perpendicular wave.

E-1: MATHEMATICS—LOGIC—THEORETICAL MATHEMATICS

364
A Ring Model for the Study of Multiplication for Complement Codes, by H. L. Garner (Univ. of Michigan); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 25–30; March, 1959.

A model is presented which, when modulo addition is used, can be used to derive multiplication correction schemata for operands expressed in either the radix complement or the diminished radix complement code.

365
A Note on a Method for Generating Points Uniformly on *n*-Dimensional Spheres, by M. E. Muller (Princeton Univ. and IBM Corp.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 19–20; April, 1959.

A method for generating points uniformly on n-dimensional spheres which is based on the relative ease of generating normal deviates and on a property of the normal distribution, is described.

366
An Efficient Method for Generating Uniformly Distributed Points on the Surface of an N-Dimensional Sphere, by J. S. Hicks and R. F. Wheeling (Socony Mobil Oil Co.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 17–19; April, 1959.

An algorithm for inductively transforming points on the surface of the $(n-1)$-sphere onto the n-sphere with preservation of uniformity of distribution, is described. The method is much more efficient than starting with uniformly distributed points on the n-cube.

367
On Exponential Digital Filters, by M. Blum (Convair); *J. Assoc. for Computer Mach.*, vol. 6, pp. 283–304; April, 1959.

The theory of an exponential digital filter for estimating the derivatives of input functions is presented. The inputs are arbitrary linear combinations of $(n+1)$ known functions plus a random stationary signal and a random noise component. The solution is available as a time-varying recursion formula.

E-2: MATHEMATICS—LOGIC—SYMBOLIC LOGIC, BOOLEAN ALGEBRA, NUMBER SYSTEMS

368
Absolute Minimal Expressions of Boolean Functions, by S. Abhyankar (Cornell Univ.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 3–8; March, 1959.

A beginning is made in the hitherto unexplored problem of finding absolute minimal expressions of Boolean functions, absolute minimals being found for Boolean functions whose point set complex consists of either one or two points.

369
A Note on Syntactic Symmetry and the Manipulation of Formal Systems by Machine, by H. Gelernter (IBM Corp.); *Infor. and Control*, vol. 2, pp. 80–89; April, 1959.

In the past few years, digital-computer programs that depart from the traditional numerical computation and data processing for which these machines were conceived have become increasingly commonplace. In many of these programs, the computer is called upon to manipulate a complex formal logistic system as a tool to implement solution of the problem. The problem of efficient machine manipulation of formal systems in which the predicates display a high degree of symmetry is discussed. The solution to the problem, embodied in a theorem and a rule of syntactic symmetry, is given. The theorem is, in fact, a metatheorem concerning formal systems, and is used in the synthesis of proofs. On the other hand, the role is an invaluable aid in the search for a proof by the so-called analytic method. The set of all syntactic symmetries for a given set of formulas is constructed and displayed in a form conducive to minimum effort programming for a computer.

370
A New Concept in Computing, by R. L. Wigington (U. S. Dept. of Defense); PROC. IRE, vol. 47, pp. 516–523; April, 1959.

A new computing scheme proposed by von Neumann, in which digital information is represented by the phase of a sine wave, is described and explained. The use of the phase together with majority logic permits the realization of logic operations. Simple aggregates are given as examples. The scheme is similar to that used in the Japansese Parametron.

371
A Synthesis Technique for Minimal State Sequential Machines, by S. Ginsburg (Natl. Cash Register Co.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 13–24; March, 1959.

A synthesis technique which always yields a minimal state sequential machine satisfying a prescribed finite set of input-output sequences is presented. An application is made to the case where a given sequential machine is to be reduced, by the merging technique, to a machine having the smallest number of states possible. Numerous examples are given.

372
On the Reduction of Superfluous States in a Sequential Machine, by S. Ginsburg (Natl. Cash Register Co.); *J. Assoc. for Computing Mach.*, vol. 6, pp. 259–282; April, 1959.

The problem of finding a (sequential) machine T, having fewer states than a given machine S, and capable of performing all the operations of S, is discussed. The solution obtained by merging all possible states of S is shown to be not necessarily the optimum solution. A practical solution to a less general problem, involving certain restrictions on S, is given.

373
Extension of Moore-Shannon Model for Relay Circuits, by M. Kochen (IBM Corp.); *IBM J. Res. & Dev.*, vol. 3, pp. 169–186; April, 1959.

The Moore-Shannon model for improving the reliability of relay circuits is extended to show that the number of redundant relays required depends on the logical function of the entire circuit. Specific studies are made of AND, OR and EXCLUSIVE-OR circuits. Intermittent failures are discussed, and a specific way to improve an AND circuit, by means of a series parallel network, is demonstrated.

374
Matrix Synthesis of High-Speed Logic, by E. J. Schubert (Burroughs Corp.); *Commun. and Electronics*, no. 41 (*Trans. AIEE*, pt. I, vol. 78), pp. 4–8; March, 1959.

A method for synthesizing function matrices from logical conditions and for de-

tecting redundancy prior to designing the logical network is described. The definition of function matrices and logical operations; the expansion of logical functions in view of additional logical conditions imposed by another variable; and minimization routines are discussed.

375
A Functional Canonical Form, by H. A. Curtis (Lewis Res. Center, N.A.S.A.); *J. Assoc. for Computing Mach.*, vol. 6, pp. 245–259; April, 1959.

An approach to the problem of designing economical switching circuits by considering the underlying structural properties of the switching functions to be realized is outlined. A functional canonical form based on the "exclusive-or" operator and useful in such design is developed. Relative switching function costs are also considered.

376
On Codes for Checking Logical Operations, by W. W. Peterson (Univ. of Florida) and M. O. Rabin (Hebrew Univ., Jerusalem); *IBM J. Res. & Dev.*, vol. 3, pp. 163–68; April, 1959.

It is shown that, both for check symbol and block coding of vectors of binary digits, there is no system of checking simpler than duplications, for all non-trivial functions of two variables, except "exclusive or" and its complement. For these latter functions, group alphabets can be used, and for block coding these are the only codes which can be used.

377
A New Method of Checking the Consistency of Precedence Matrices, by R. B. Marimont (Natl. Bur. of Standards); *J. Assoc. for Computing Mach.*, vol. 6, pp. 164–171; April, 1959.

Both graphical and Boolean matrix representations of a set of precedence relations are discussed and criteria for consistency are enunciated. It is shown that, graphically, the existence of a closed loop of directed links is equivalent to inconsistency, while a precedence matrix is consistent if and only if every principal submatrix has at least one zero row or zero column.

378
Decoding Nets and the Theory of Graphs, by Z. Pawlak (Warsaw Polytech. Inst.); *J. Soc. for Industrial and Appl. Math.*, vol. 7, pp. 1–5; March, 1959.

The problem of minimizing decoding nets may be expressed in the language of the theory of graphs. The necessary condition that a decoding net has the least number of diodes is given.

E-3: MATHEMATICS—LOGIC—NUMERICAL ANALYSIS

379
A Mathematical Procedure for Machine Division, by R. E. Gilman (Brown Univ.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 10–12; April, 1959.

A division procedure applicable to machines which use the principle of combining, separately, right- and left-hand components is described. Both divisor and dividend are adjusted iteratively by multiplications that make the divisor approach 0.9. Several dividends may be divided simultaneously by the same divisor. In general, n-operations produce 2^n place accuracy.

380
Transposing Matrices in a Digital Computer, by R. F. Windley (Univ. of Leeds); *Computer J.*, vol. 2, pp. 47–48; April, 1959.

A technique for transposing matrices that is very economical of storage space is described. The transposition is considered as a permutation which is broken up into its component cycles.

381
A Procedure for the Diagonalization of Normal Matrices, by H. H. Goldstine (Princeton Univ.), and L. P. Horwitz (IBM Corp.); *J. Assoc. for Computing Mach.*, vol. 6, pp. 176–195; April, 1959.

The Jacobi procedure for symmetric matrices (based on tri-diagonalization) is extended to normal matrices. The technique consists in iteratively minimizing the sum of the squares of the off-diagonal elements. An iterative procedure is shown to be possible, even for the case where the lowest value for the change is non-negative.

382
Computation of Sin N, Cos N and $\sqrt[m]{N}$ using an Electronic Computer, by E. G. Kogbetliantz. (IBM Corp.); *IBM J. Res. & Dev.*, vol. 3, pp. 147–152; April, 1959.

Rational Padé approximations to a power series are of the form $P_M(x)/Q_N(x)$, where $P_M(x)$ and $Q_N(x)$ are polynomials. The theory is applied to sin N, cos N and $\sqrt[m]{N}$, and computer routines are outlined for each case. The method is flexible and gives any desired accuracy with fewer machine operations than other methods. For higher-order radicals, Newton's method is too slow, and Padé approximations provide more efficient routines.

383
A Note on the Downhill Method, by G. C. Caldwell (North Carolina State College); *J. Assoc. for Computing Mach.*, vol. 6, pp. 223–225; April, 1959.

A method for computing successive increments in x and y in the so-called "downhill" method for solving the equation $f(z)=0$, where $f(z)$ is analytic, is given. The "downhill" method is suitable for computers in that it always converges and is programmable for the general case.

384
Stability of Numerical Solution of Differential Equations, by W. E. Milne and R. R. Reynolds (Oregon State College); *J. Assoc. for Computing Mach.*, vol. 6, pp. 196–203; April, 1959.

The instability of Milne's method of solving ordinary differential equations is due to the use of Simpson's rule for the final integration. It is shown that occasional application of Newton's "three-eighths" quadrature formula effectively damps out the error oscillation without detriment to the final solution.

385
Numerical Solution of Laplace's Equation—Given Cauchy Conditions, by I. Sugai (IBM Corp.); *IBM J. Res. & Dev.*, vol. 3, pp. 187–188; April, 1959.

With Cauchy boundary conditions Laplace's equation is unstable, in that exponential growth of errors occurs during numerical analysis by finite difference methods. A "rule-of-thumb" method for estimating the magnitude of propagated errors and thus choosing the mesh size is described.

386
Elimination of Special Functions from Differential Equations, by J. E. Powers (Univ. of Oklahoma); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 3–4; March, 1959.

It is pointed out that common functions, such as trigonometric or exponential functions, which satisfy simple differential equations can be eliminated from a set of ordinary differential equations by considering an equivalent expanded set of equations which do not contain the functions. Solution of the new set of equations either by Runge-Kutta techniques or by analog computer is often simpler than solution of the original set.

387
Boundary Contraction Solution of Laplace's Differential Equation, by H. W. Milnes (G. M. Corp.) and R. B. Potts (Univ. of Toronto); *J. Assoc. for Computing Mach.*, vol. 6, pp. 226–235; April, 1959.

A numerical solution of the Dirichlet problem within a circle by successive contraction of the boundary (on which the solution is known) is described. Appropriate approximating relations are presented and their stability discussed. The method is much more economical of storage space than the classical finite-difference equation procedure.

388
A Method of Normalized Block Iteration, by E. H. Cuthill (U. S. Navy) and R. S. Varga (Westinghouse Elec. Corp.); *J. Assoc. for Computing Mach.*, vol. 6, pp. 236–244; April, 1959.

In solving elliptic partial-differential equations, successive block over-relaxation is known to converge faster than point over-relaxation. A method by which the block scheme may be performed for the *same* number of arithmetical operations per iteration as the point scheme is given.

389
An Iterative Method for Fitting the Logistic Curve, by J. R. Howell (Univ. of Dayton); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 5–6; March, 1959.

An iterative method for finding the best least squares fit to the logistic curve $p(x, \alpha, \beta)=1/(1+e^{\alpha-\beta x})$ is given. An arbitrary multiplier θ is applied to the corrections α, β at each stage to ensure convergence.

390
Monte Carlo Solutions of Boundary Value Problems Involving the Difference Analogue of $\partial^2u/\partial x^2+\partial^2u/\partial y^2+(K/y)(\partial u/\partial y)=0$, by L. W. Ehrlich (Ramo-Wooldridge Corp.); *J. Assoc. for Computing Mach.*, vol. 6, pp. 204–218; April, 1959.

Monte Carlo techniques are applied to various elliptic partial-differential equations, and the results are compared with iterative techniques. The rate of convergence is slow, and the error can be shown to be independ-

ent of dimension. If the solution is desired for only one point, the Monte Carlo technique may well be competitive.

391
Numerical Quadrature in Many Dimensions, by D. Morrison (Space Tech. Labs.); *J. Assoc. for Computing Mach.*, vol. 6, pp. 219–222; April, 1959.

A technique for evaluating a class of multiple integrals by reducing the problem to one of approximating the integrals of functions with fewer variables, is described.

392
On Computing Radiation Integrals, by R. C. Hansen (Hughes Aircraft Co.), L. L. Bailin (Univ. of Southern California), and R. W. Rutishauser (Litton Industries, Inc.); *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 28–31; February, 1959.

The relative merit and cost of four ways of evaluating typical radiation integrals containing spherical Bessel functions are investigated. These methods are desk machine evaluation of a finite series, integration of the appropriate differential equation by a Reeves Electronic Analog Computer and by a Litton 40 Digital Differential Analyzer, and numerical integration on the IBM 704 computer. Results are generally applicable to equations separated from a Helmholtz or wave equation.

E-5: MATHEMATICS—LOGIC—INFORMATION THEORY

393
Some Cyclic Error-Correcting Codes with Simple Decoding Algorithms, by E. Prangle (A. F. Cambridge Res. Center); *U. S. Gov. Res. Repts.*, vol. 31, p. 105(A), February 13, 1959; PB 135 504.

The properties of two cyclic error-correcting codes of length 21 in the symbols 0 and 1 are discussed. The codes are of the orders 2^{11} and 2^{12} respectively. In each case, coding is by means of a linear recursion. Both codes have simple decoding algorithms. The second is quasi-perfect.

394
The Morse Distribution, by M. Friemer, B. Gold and A. L. Tritter (M.I.T. Lincoln Lab.); IRE TRANS. ON INFORMATION THEORY, vol. IT-5, pp. 25–31; March, 1959.

A problem which arose during research involved in designing a machine to translate hand-keyed Morse Code into printed text is discussed. The problem may be stated as follows: Let $X=\{x_i: i=1, 2, \cdots, n\}$ be a sequence of independent random variables all of which have the same distribution. Assume that the probability that $x_i=x_j$, $i\neq j$, is zero. Let k be a positive integer $\leq n$, and consider all subsequences x_i, $x_{i+1}, \cdots, x_{i+k-1}$ of X consisting of k consecutive variables. Let us distinguish, with a check ($\surd$), the largest member of each such subsequence. The probability that exactly r members of the sequence X are not checked has been studied and partially tabulated.

395
Full Decodable Code-Word Sets, by M. P. Schützenberger (Faculté des Sciences de Poitiers, France) and R. S. Marcus (Mass. Inct. Tech.); IRE TRANS. ON INFORMATION THEORY, vol. IT-5, pp. 12–15; March, 1959.

The restrictions imposed on a set of code words by the decodability condition is considered. A generating function is defined that describes the composition of the code words. The relation between the generating function and a "full" set of code words is found. This relation shows that the sum of arbitrary probabilities associated with the words of a full set must be one. A full set of code words is one to which no code word can be added and still keep the set decodable. It is also shown that a full set is "completable." For a completable set of code words any string of symbols can be made into a sentence by adding a suitable prefix and a suffix.

396
On a Property of Wiener Filters, by M. Zakai (Ministry of Defense, Israel); IRE TRANS. ON INFORMATION THEORY, vol. IT-5, pp. 15–17; March, 1959.

Let $Y(\omega, \alpha)$ be the Wiener filter designed to yield an output which is the least-square approximation to $s(t+\alpha)$ where $s(t)$ is the desired signal input. Let $Y_L(\omega)$ be the Wiener filter designed to yield an output which is the least-square approximation to some linear operation L on the desired input signal. The following simple relationship has been shown to hold between $Y(\omega, \alpha)$ and $Y_L(\omega)$: if $s(t)$ is the desired input signal and $L_\alpha[s(t+\alpha)]$ is the desired output, where L_α is some linear operation with respect to α, then $Y_L(\omega)=L_\alpha[Y(\omega, \alpha)]$.

F: PERSONNEL

397
1958 PGEC Membership Survey Report, by K. W. Uncapher (Rand Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 60–67; March, 1959.

In order to sense the PGEC's current position in the computing industry and in order for each member to assess his position within the Group, a second membership survey was completed in the Fall of 1958. The results of this survey are reported herein. Tabular data, bar graphs, and salary curves are included to aid in assessment of the Group and an individual's position in the industry, particularly with respect to education, experience, field of endeavor, salary and geographical location.

398
Positions for High School Graduates in Electronic Data Processing, by E. J. Haga (Sacramento State College); *Computers and Automation*, vol. 8, pp. 13–16; February, 1959.

A survey made to determine the positions in data-processing which are open to high-school graduates is discussed. Operating, maintenance, and clerical jobs are open to such students, though those with superior ability and experience would qualify as programmers and systems analysts. In general, ability, experience and personal characteristics are more important than formal education in this field.

399
Automation in the Post Office; *Computer Bull.*, vol. 2, pp. 78–79; February–March, 1959.

Details of the staff consultation employed by the British Post Office to facilitate the introduction of computer operations are given.

I: STANDARDS

400
IRE Standards on Static Magnetic Storage: Definitions of Terms, 1959; PROC. IRE, vol. 47, pp. 427–430; March, 1959.

Standardized meanings for static magnetic storage terms are given.

J: SUMMARIES AND REVIEWS

401
A Review of the Electronic Computer Exhibition and Business Symposium, by H. W. Gearing (The Metal Box Co. Ltd.) and D. W. Hooper (Natl. Coal Board); *Computer Bull.*, vol. 2, pp. 71–76; February–March, 1959.

The papers presented at the British Computer Exhibition and Business Symposium held in London in November–December, 1958, are reviewed, and brief comments given on the computer equipment on display.

402
Programming Services and Advice for Prospective Computer Users and Others, by F. C. dePaula; *Computer Bull.*, vol. 2, pp. 87–91; April–May, 1959.

The situations in which it is convenient to have programming services and/or machine time are reviewed. Different sources of such services and advice are listed. The problems met by those providing such services and computers on a service basis are discussed.

403
A Survey of British Digital Computers, by J. L. F. de Kerf (Gevaert Photo-Producten, Belgium); *Computers and Automation*, vol. 8, pp. 25–26, 29, March, 1959; pp. 34–36, April, 1959; pp. 38–39, May, 1959.

Brief descriptions of the leading British digital computers on the market are given.

404
Survey of Progress and Trend of Development and Use of Automatic Data Processing in Business and Management Control Systems of the Federal Government, as of December 1957; *Commun. Assoc. for Computing Mach.*, vol. 2, pp. 22–26; April, 1959.

The present utilization of computers by the Federal Government and the future trend of development are reviewed. Check-accounting in the Department of the Treasury, supply and logistics in the Department of Defense, and census applications are among the items discussed.

405
Computer Power: a Public Utility? by A. O. Mann (Philco Corp.); *Computers and Automation*, vol. 8, pp. 11–16; April, 1959.

A close analogy in regard to use, cost, peak load, and excess capacity is drawn between computer installations and electric power systems. The suggestion is put forward that in order to make full use of the capabilities of computing systems, they should evolve into public utilities.

SUBJECT INDEX

AUTHOR INDEX

JOINT COMPUTER COMMITTEE

SENEWS

SCIENCE EDUCATION SUBCOMMITTEE NEWSLETTER

Vol. 2, No. 3 September, 1959

SENEWS is published as an integral part of Communications of the ACM and the IRE TRANSACTIONS ON ELECTRONIC COMPUTERS; in addition to this, the AIEE makes a separate distribution to their computer group.

At the present time, teachers will be placed on a permanent distribution list at no charge by so notifying the editor. We cannot guarantee that this will be a permanent arrangement, but it should get *SENEWS* to teachers in the easiest manner.

With school about to open and the new year ahead of us, we would like to initiate another change in policy for *SENEWS*. Henceforth, we will welcome and publish, if of sufficient quality, technical articles or reports on projects or organizations *written by* secondary school students. We hope this will provide a place where a student can publish his work on a national scale. Each paper will receive careful review by a highly qualified member of the computing profession, but the work will not have to compete with professionals in the field. No editorial policy has been established yet, and probably none will be until some student papers have been submitted. We would like to encourage students to do more writing about their work and, therefore, we urge *SENEWS* readers to tell their students and science club members of this opportunity.

Because of the greatly enlarged circulation and requests for this particular article, we are reprinting Rollin Mayer's "PAPAC-00, A Do-It-Yourself Paper Computer." Reprints of this and other articles are listed on page 428 and are available upon request.

MICHAEL WARSHAW, *Chairman*
JCC Science Education Subcommittee
The RAND Corporation
Santa Monica, Calif.

COMPUTING CURRICULUM AT CONCORD HIGH SCHOOL

Concord, Massachusetts, High School Mathematics Department will conduct a streamlined repetition of the spring term unit on computers as reported in *SENEWS* Vol. I, No. 1 (July, 1958). A three-week curriculum unit for twelve students is being designed by Norton Levy, Concord High School Mathematics Department, assisted by members of M.I.T. Lincoln Laboratory staff. The engineers act as consultants, arrange computer facilities tours, and furnish a demonstration-lecture team for a major part of the course. The unit will include number systems, simple programming theory, homework programs (a trial run during a visit to Lincoln's TX-2 computer is planned), computer applications, and a brief study of computer logic.

We hope to get across to the students the idea that a wide range of problems are such that data processing machines, such as TX-2, can materially aid in their solution. It is considered important that the student learn the respective roles of mathematics, machine, and programmer.

Two programs will be written and demonstrated on TX-2. The first will be a polynomial display—probably a cubic. Since we are using fixed-point arithmetic, the problems of scaling to avoid overflow and other limitations will show up quickly. We intend to start with a simple though limited program, point out its limitations, and show how the programmer can change his program to extend its scope. Most of the programming will be done by the class, at school and in homework. Some of the work, especially the IN-OUT portion, will be done by the teachers.

The second program, a simulated moon-rocket display, will consist of showing three spots on the display screen: one representing the earth, being fixed in the center; the second representing the moon, and traveling around in about half a minute; the third representing the rocket, whose motion will be calculated based on the gravitational fields of the earth and moon. The simulated rocket engine will be controlled directly from the screen by means of a photocell, or by preset parameters which will demonstrate successful lunar orbits. Study of this program will illustrate methods of handling simple differential equations and methods of communication between operator and machine (*e.g.*, via the photocell). Operation of the program will illustrate some of the dif-

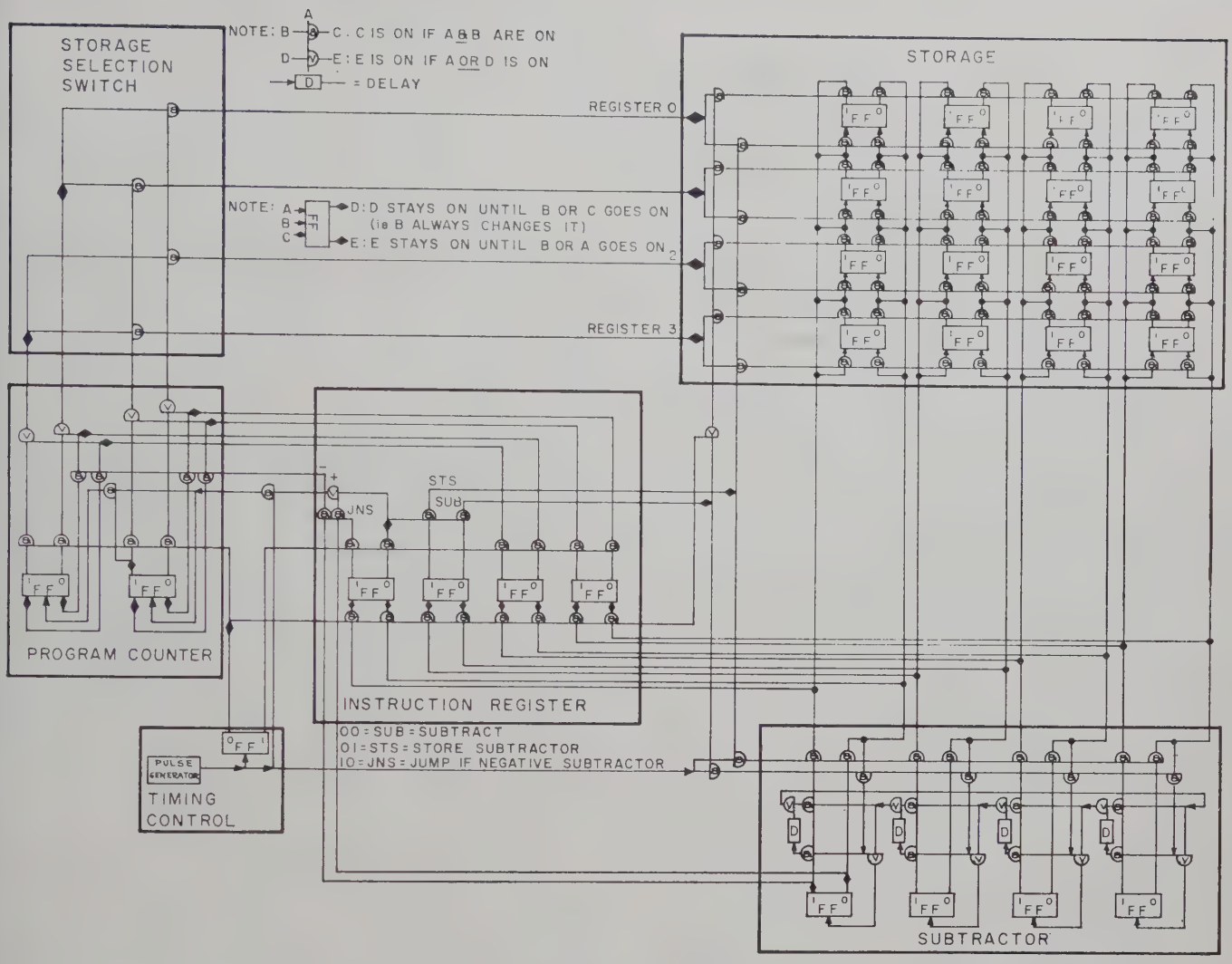


Fig. 1—A general-purpose computer.

ficulties involved in shooting the moon, and will convey a feeling for the behavior of rockets in space.

The study of computer logic will include a discussion of Fig. 1. This computer has only four storage registers, but it is easy to describe how to add as many as desired. It has no input or output equipment, but lights and switches (or other equipment) can be directly connected to the Flip-Flops (FF) of any storage register; it has only three instructions, but the addition of a few more can easily be described. The discussion will include methods of programming any other instruction using only the three instructions shown (*e.g.*, clear and subtract x=STS k, SUB k, SUB x), thus qualifying this computer for the title "general-purpose." Briefly, the computer works as follows: Every instruction takes two steps as controlled by timing control. On the first step, the instruction is obtained from the storage register specified by the program counter, and is placed in the instruction register. On the second step, this instruction is performed. If it is a JNS (Jump if Negative Subtractor), and the subtractor is negative, then the program counter is changed to agree with the right half of the instruction; but if the subtractor is positive, then the program counter is merely increased by "one," as it is on both of the other instructions. If the instruction is STS (Storage Subtractor) then the number in the subtractor is copied into the storage register specified by the instruction. If the instruction is SUB (Subtract) then the negative (in "1's complement" form) of the number in the specified storage register is added to the subtractor.

This curriculum experiment guided by a teacher-engineer team is sponsored by the Ford Foundation's School and University Program for Research and Development, under grant to Harvard's School of Education.

N. Levy
Math Dept., Concord High School
Concord, Mass.
R. Mayer
Mitre Corp.
Lexington, Mass.
Formerly with M.I.T., Lincoln Lab.
H. Vanderburgh
M.I.T. Lincoln Lab.
Lexington, Mass.

A Computing Course in San Antonio

In September, 1958, the San Antonio Chapter of the University of Texas Ex-Students Association, decided that it might be worthwhile to provide an "Introductory Course to Digital Computers." It was felt that the best approach would be to have a course for high-school teachers first and then if the interest by the students was sufficient a course would be developed for students. The San Antonio Independent School District provided the names of high-school mathematics teachers in the District, and a total of seventy-seven letters were sent to these teachers describing the purpose of the computer course. Thirty-four teachers responded and stated they had an interest in attending; however, there were only eighteen of these who could meet on the same evening.

The course met once a week for ten weeks for a total of twenty hours. Dr. David Young, Director of the University of Texas Computation Center, reserved one entire Saturday for the teachers' use of the IBM 650 computer at the University of Texas Computation Center. The purpose of the course was to give a wide coverage of the computer field rather than concentrate on many details. The first three meetings consisted primarily of a discussion of the history of computers and a brief description of the chronological development of the more modern machines. The basic punch card machines were covered in one two-hour session and the general characteristics of computers were covered in another two-hour session. The last five meetings were devoted to the discussion of problem analysis, programming and coding, with the last three meetings devoted to programming for the IBM 650 computer. Throughout the entire ten meetings various topics of mathematics related to computer usage were discussed and considerable time was devoted to bringing the math teachers up to date on techniques of automatic coding, symbolic programming, and other techniques in communicating with computers. It is felt that the teachers received a wide background of information that would enable them to pursue the computer field on their own.

The teachers stimulated sufficient interest in their students to have a students' course which was begun in March of this year. Approximately sixty students attended the first meeting. The number of students stabilized to about thirty-five as the course progressed through ten meetings. The course for the students consisted of essentially the same material as was presented to the teachers; however, the history portion was covered more rapidly and more time was devoted to discussion of interesting mathematical topics related to computers. Both the teacher and student groups visited the IBM 705 installation at Kelly Air Force Base, and the IBM 650 Tape System at Lackland Air Force Base.

It is planned to continue this program during the coming school year since there are many school districts in the San Antonio area that were not included in the first course.

J. H. Ward, Jr.
Wright Air Dev. Center
Lackland Air Force Base, Tex.

PAPAC-00, A Do-It-Yourself Paper Computer

In less than an hour you can build the simplified digital computer shown in Fig. 1, using only a pair of scissors, three dozen common pins, and the parts shown in Figs. 1 and 2. This computer was developed from the model demonstrated in the Concord, Massachusetts, High School lectures on computers reported in *SENEWS*, Vol. I, No. 1.

From the discussion below, the computer expert will recognize that "PAPAC double zero" contains most of the units of a large-scale computer, but in simplified form. The *control unit* includes a counter and a system for controlling the parts of the computer according to the instruction being performed (in this model a simple fixed instruction is used; a large computer can draw from several instructions obtained from storage). The *storage unit* includes registers, bus, and selection switch; register contents are changed by hand rather than by the computer. The *arithmetic unit* can add. *Input and output units* have been eliminated by allowing the operator to deal with the insides of the computer directly rather than by way of complicated equipment. Proprietary rights are held by the author.

In *operation*, PAPAC-00 follows the same fixed instruction over and over again. This instruction is: "Read the number out of the currently-selected storage register and add it to the adder, then get ready to use the next storage register for the next time." The "Counter" keeps track of which storage register to use next; since there are only two registers, numbered "0" and "1," the counter alternates between them. The "Switch" is controlled by the counter and allows only the selected register to be operated. Each "storage register" contains only a single binary "cell"; when the register is operated, the cell is forced against the "Bus" if the cell is set to "1." If a "1" has been read out in this way, the bus actuates the "adder," preparing it to add the "1." If the cell is set to "0," the bus and adder are not operated, and "0" is added to the adder. Binary sums are as follows: $0+0=0$, $0+1=1$, $1+0=1$, $1+1=10$. The adder forms these sums correctly except that in the last case it forms a sum of "0" because it can handle only one digit. The "Control," pushed back and forth by hand, performs this fixed instruction by operating the counter and switch, and by returning the bus to its "0" position (if it had read out a "1") causing the sum to be formed in the adder.

To *assemble* PAPAC-00, Fig. 1 should be used as the base, and the shapes of Fig. 2 should be fitted over it by following these steps:

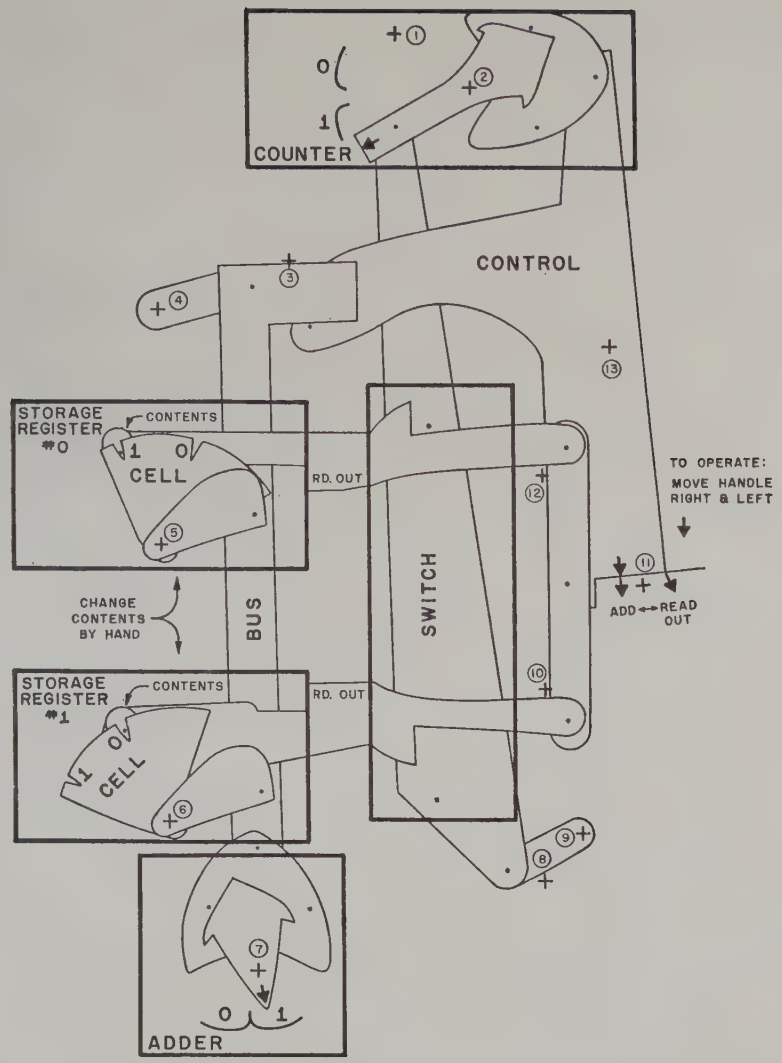


Fig. 1—PAPAC-00, a two-register, one-bit, fixed-instruction binary digital computer.

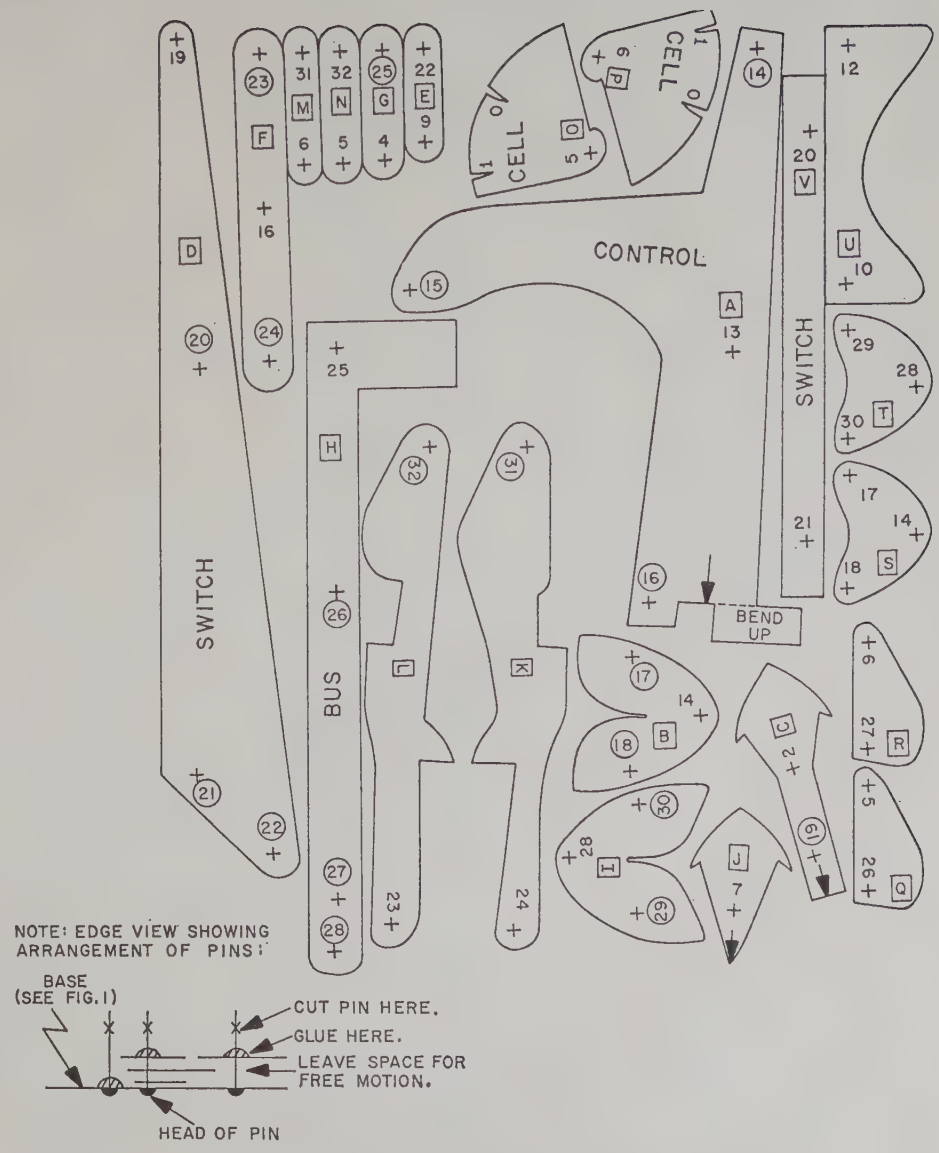


Fig. 2—Parts for PAPAC-00.

1) Punch a pinhole exactly through the intersection of each cross (+) in Figs. 1 and 2 (but not the dots in Fig. 1).

2) Cut out exactly on the lines, the parts in Fig. 2, in any order. They are marked with a letter in a square box, from [A] to [V], and the next steps will be easier if you place each piece on the table in alphabetic order as you cut it out.

3) Place a pin up through each hole with a circle number (from (1) to (32)).

4) Taking each part of Fig. 2 in alphabetic order, place its *un*circled number holes down over the correspondingly numbered pins.

5) In first operating the computer you may find that some parts jam because the upper piece is down too far on the pins; pry such pieces up a little to provide space for free motion.

6) The construction can be refined by cutting the pins and gluing the uppermost part to the remaining length.

Caution:

a) Don't cut the stop pins too short.

b) Glue only *one* moving part to the same pin.

ROLLIN P. MAYER
Mitre Corp.
Lexington, Mass.
Formerly with M.I.T. Lincoln Lab.

JCC SCIENCE EDUCATION LITERATURE

Projects

[1] Decimal to Binary Converter No. 1 (Schematic only).
[2] Decimal to Binary Converter No. 2 (Schematic only).
[3] A Paper Computer—PAPAC-00—R. P. Mayer

Articles

[1] M. Warshaw, "How computers work."
[2] G. E. Forsythe, "Bibliography on High School Mathematics Education."
[3] W. G. Schmidt, "Boolean Algebra and the Digital Computer."

PGEC News and Notices

EJCC at Boston December 1–3

Plans are near completion for the 1959 Eastern Joint Computer Conference, to be held in the Statler Hotel, Boston, Mass., from Tuesday, December 1, to Thursday, December 3. Six technical sessions (no parallel sessions) will include papers on Larc, Stretch, Project Lighting, tunnel diodes, character recognition, systems design, circuits, and programming. On Wednesday evening, December 2, there will be four (parallel) panel discussion sessions:

"Systems Aspect of the Utilization of Kilomegacycle Components,"

"Judicious Use of Your Computer,"

"The Role of the Computer in the Engineering Design of Computers,"

"Large-Signal Equivalents in the Analysis of Circuit Tolerance."

Two trips are planned, to Lincoln Laboratories and to Arthur D. Little, Inc. Arrangements have also been made for a cocktail party on December 1, and a dinner on December 3, at which Willis Ware will present highlights of the recent trip of NJCC representatives to the Soviet Union, and will answer questions from the audience. A $300 award for the best presentation of a significant technical contribution will also be made at the dinner.

For a listing of the complete program of the Conference, see the November issue of Proceedings of the IRE.

Sixth Annual Computer Applications Symposium

On October 28 and 29, the Sixth Annual Computer Applications Symposium will be held in Chicago, under the sponsorship of Armour Research Foundation. Detailed information can be obtained from R. B. Wise, Program Chairman, Armour Research Foundation, 10 West 35 St., Chicago 16, Ill.

Chicago Host to NEC

The fifteenth annual National Electronics Conference was held Monday through Wednesday, October 12–14, at the Sherman Hotel in Chicago. Sessions of interest to computer people included:

Monday—"Adaptive Servomechanisms," "Computers," "Perception and Recognition."

Tuesday—"Numerical Analysis and Switching Theory," "Solid-State Circuits."

Wednesday—"Systems Analysis."

Call for Papers—Solid-State Circuits Conference

The Seventh Annual Solid-State Circuits Conference will be held on February 10–12, 1960 in Philadelphia, Pa. The conference, sponsored jointly by the IRE, the AIEE, and the University of Pennsylvania, will feature papers on circuit properties, circuit philosophy, and design techniques related to solid-state devices in the following areas:

1) Microcircuit techniques for improved speed or utilization of volume, weight, cost, and increased reliability.

2) Solid-state devices performing an integrated or alterable circuit function.

3) Significant contribution to art in flexibility, bandwidth, gain, stability, reliability, etc.

4) Solid-state memory, storage and logic devices such as thin films, multiaperture magnetics, twistors, optoelectrics.

5) Significant contributions to solid-state microwave electronics such as parametric amplifiers, masers, parametrons.

6) Low-temperature solid-state electronics for memory, logic, *i.e.*, cryoelectrics.

Deadline for abstracts of 300 or more words has been set for October 9. Program Chairman is Tudor R. Finch, Bell Telephone Laboratories, Inc., Murray Hill, N. J.

Challenge of Computers Theme of 1960 WJCC

"Computers—Challenge of the Next Decade" will be the theme of the 1960 Western Joint Computer Conference, May 3–5, in San Francisco, Calif.

Special emphasis will be placed on areas in which growth and development of are anticipated during 1960–1970. These include language translation, data retrieval, and self-teaching systems.

To facilitate comprehension of the technical sessions, the proceedings of the Conference will be distributed at the registration desks. The Technical Program Committee has requested that three copies of an original draft of the complete paper be submitted by November 9. Authors will be notified of acceptance by January 25, 1960, and a final copy of the paper will be required by March 1 for publication in the proceedings. Committee Chairman is H. M. Zeidler, Stanford Research Institute, Menlo Park, Calif.

Task Force on Data Systems Languages Invites Members

The Intermediate-Range Task Force of the committee on Data System Languages has invited the participation as associate members of people engaged in all aspects of business data processing applications. The particular mission of the IRTF is to specify a common language for describing data processing systems. Associate members will be recipients of the publications of the Task Force and will, from time to time, be called upon to respond to surveys, give opinions, or otherwise guide the thinking of the working force.

For further information, write to A. E. Smith, Chairman, Intermediate-Range Task Force, Code 280, Navy Department, Washington 25, D. C., and include a statement of interest and experience.

INFORMATION FOR AUTHORS

IRE TRANSACTIONS ON ELECTRONIC COMPUTERS is published quarterly, in March, June, September, and December, with a distribution of over 8500 copies, largely to engineers, logicians, and supervisors in the computer field. Its scope includes the design, theory, and practice of electronic computers and data-processing machines, digital and analog, and parts of certain related disciplines such as switching theory and pulse circuits.

If a paper of widespread interest beyond the computer field is submitted, it will be recommended to the Editor of PROCEEDINGS OF THE IRE for publication. If our reviewers feel that a paper should be submitted to a different IRE TRANSACTIONS, we will so recommend to the author.

Publication time in IRE TRANSACTIONS ON ELECTRONIC COMPUTERS, from receipt of the original manuscript to mailing of the issue, is normally in excess of five months, but can be made as little as $3\frac{1}{2}$ months if the occasion demands and the manuscript is carefully prepared.

To avoid delay, please be guided by the following suggestions:

A. Process for Submission of a Technical Paper

1) Send to the Editor three copies of your manuscript, each copy complete with illustrations. (For Letters to the Editor, two copies will do.)

2) Enclose originals for the illustrations, in the style described below. Alternatively, be ready to send the originals immediately upon acceptance of the paper.

3) Enclose a separate sheet giving your preferred address for correspondence and return of proofs.

4) Enclose a technical biography and photograph of each author, or be ready to supply these upon acceptance of the paper. For biography style, see any IRE journal.

5) If the manuscript has been presented, published, or submitted for publication elsewhere, please so inform the Editor. Our primary objective is to publish technical material not available elsewhere, but on occasion we publish papers of unusual merit that have appeared or will appear before other audiences.

B. Style for Manuscript

1) Typewrite, double or $1\frac{1}{2}$ space; use one side of sheet only. (Good office-duplicated copies are acceptable.)

2) Provide an informative 100- to 250-word summary (abstract) at the head of the manuscript. It will appear with the paper and also separately in PROCEEDINGS OF THE IRE.

3) Provide a separate double-spaced sheet listing all footnotes, beginning with "*Manuscript received by the PGEC _____," and "†(Affiliation of author)," and continuing with numbered references. Acknowledgment of financial support is often placed at the end of the asterisk footnote.

4) Give complete references, insofar as possible. See footnotes in this or previous issues for examples of IRE style. References will be printed as footnotes in the column where first mentioned.

5) You may choose to provide a "Bibliography" at the end of the paper, with items referred to by a numeral in square brackets, *e.g.*, [12], to supplement or supplant footnote references.

6) Provide a separate sheet listing all figure captions, in proper style for the typesetter, *e.g.*: "Fig. 1—Example of a disjoint and distraught manifold."

C. Style for Illustrations

1) Originals for illustrations should be sharp, noise-free, and of good contrast. We regret that we cannot provide drafting or art service.

2) Line drawings should be in India ink on drafting cloth, paper, or board. Use $8\frac{1}{2} \times 11$ inch size sheets if possible, to simplify handling of the manuscript.

3) On graphs, show only the coordinate axes, or at most the major grid lines, to avoid a dense, hard-to-read result.

4) All lettering should be large enough to permit legible reduction of the figure to column width, perhaps as much as 4:1.

5) Photographs should be glossy prints, of good contrast and gradation, and any reasonable size.

6) Number each original on the back, or at the bottom of the front.

7) Note item B-6 above. Captions lettered on figures will be blocked out in reproduction, in favor of typeset captions.

Mail all manuscripts to:

Dr. Howard E. Tompkins, *Editor*
IRE TRANSACTIONS ON ELECTRONIC COMPUTERS,
The Moore School of E.E., University of Pennsylvania
200 South 33rd Street
Philadelphia 4, Pa.